NASA Contractor Report 189128

NASA-CR-189128
19920011949

Fluid Mechanics Experiments in Oscillatory Flow

Volume II–Tabulated Data

J. Seume, G. Friedman, and T.W. Simon
University of Minnesota
Minneapolis, Minnesota

March 1992

Prepared for
Lewis Research Center
Under Grant NAG3–598

NASA
National Aeronautics and
Space Administration

ABSTRACT

Results of a fluid mechanics measurement program in oscillating flow within a circular duct are presented. The program began with a survey of transition behavior over a range of oscillation frequency and magnitude and continued with a detailed study at a single operating point. Such measurements were made in support of Stirling engine development. Values of three dimensionless parameters, Re_{max}, Re_w, and A_R, embody the velocity amplitude, frequency of oscillation and mean fluid displacement of the cycle, respectively. Measurements were first made over a range of these parameters which included operating points of all Stirling engines. Next, a case was studied with values of these parameters that are representative of the heat exchanger tubes in the heater section of NASA's Stirling cycle Space Power Research Engine (SPRE). Measurements were taken of the axial and radial components of ensemble-averaged velocity and rms-velocity fluctuation and the dominant Reynolds shear stress, at various radial positions for each of four axial stations. In each run, transition from laminar to turbulent flow, and its reverse, were identified and sufficient data was gathered to propose the transition mechanism. Models of laminar and turbulent boundary layers were used to process the data into wall coordinates and to evaluate skin friction coefficients. Such data aids in validating computational models and is useful in comparing oscillatory flow characteristics to those of fully-developed steady flow.

Data were taken with a contoured entry to each end of the test section and with flush square inlets so that the effects of test section inlet geometry on transition and turbulence are documented.

The following is presented in two volumes. Volume I contains the text of the report including figures and supporting appendices. Volume II contains data reduction program listings and tabulated data (including its graphical presentation).

ACKNOWLEDGEMENTS

The following work was sponsored by the Lewis Research Center of NASA under grant NASA/NAG3-598. The authors thank the grant monitors, James Dudenhoefer and Roy Tew for their guidance.

Consultation on instrumentation and 3D graphics was provided by Robert Hain, the comparison of data to steady-flow correlations in this report was done by Terry Johnson and much of the document preparation was with the aid of Phillip Tuma and Amy Johnson.

CONTENTS VOLUME I

CONTENTS VOLUME II

NOMENCLATURE

Symbol	Units	Explanation
$A_R = \frac{2x_{m,max}}{l}$		Relative amplitude of fluid displacement
A^+		Empirical effective sublayer thickness for Van Driest model
$c_f = \frac{2u_*^2}{u_m^2}$		Skin-friction coefficient
D	m	Piston diameter
d	m	Duct inner diameter
f	sec^{-1}	Frequency
k_T		Correction factor for tangential cooling of hot-wire sensor
l	m	Duct length
P	bar	Fluid static pressure
p^+		Pressure gradient parameter
$\bar{p}$	bar	Average pressure
r	m	Cross-stream coordinate, measured from the duct centerline
R	m	Pipe inner radius
$Re_d = \frac{u_m d}{n}$		Reynolds number for steady flow, based on the duct diameter and the bulk-mean velocity
$Re_{max} = \frac{u_{m,max} d}{\nu}$		Reynolds number based on the duct diameter and the amplitude of the bulk-mean velocity
$Re_\omega = \frac{\omega d^2}{4\nu_o}$		Kinetic Reynolds number, or Valensi number
$Re_\delta = u_{m,max}\,\delta/\nu$		Reynolds number based opon Stokes layer thickness
$Re_{\delta_2} = \delta_2 u_m/\nu 2$		Momentum thickness Reynolds number

Symbol	Units	Explanation
s	m	Streamwise distance measured from open end of duct, or the stroke of the piston
$Str = \frac{\omega d}{u_{m,,max}} = \frac{4Va}{Re_{max}}$		Strouhal Number
t	sec	Time
T	℃	Fluid temperature
u	m/sec	Instantaneous velocity
$\overline{u}$	m/sec	Streamwise component of ensemble-averaged velocity
$u' = \sqrt{\overline{u'^2}}$	m/sec	Streamwise component of rms-velocity fluctuation
u_{eff}	m/sec	Effective cooling velocity
u_∞	m/sec	Freestream velocity
u_m	m/sec	Bulk-mean velocity
$u_{m,max}$	m/sec	Amplitude of the bulk-mean velocity
u_n	m/sec	Velocity component normal to sensor
u_T	m/sec	Velocity component tangential to sensor
$u_* = \sqrt{\frac{\tau_w}{\rho}}$	m/sec	Friction velocity
$u^+ = \frac{u}{u_*}$		Nondimensional velocity, in wall coordinates
$\overline{v}$	m/sec	Radial component of ensemble-averaged velocity
v'	m/sec	Radial component of rms-velocity fluctuation
V	volts	Transducer voltage
$Va = \frac{\omega d^2}{4\nu}$		Valensi number
$-\overline{u'v'}$	m^2/sec^2	Reynolds shear stress
x	m	Streamwise distance, measured from drive end of duct

Symbol	Units	Explanation
x_m	m	Amplitude of displacement of bulk fluid
y	m	Cross-stream coordinate, measured from the duct wall
$y^+ = \frac{yu_*}{n}$		Distance normal to the wall in inner coordinates

Greek

$\alpha = \sqrt{Va}$		Womersely parameter
$\delta = (2\nu / \omega)^{1/2}$	m	Stokes-layer thickness
δ_2	m	Momentum thickness
θ	(°)	Crank angle within the cycle
κ		Karman constant
μ	N•sec/m^2	Dynamic viscosity
ν	m^2/sec	Kinematic viscosity
ν_o	m^2/sec	Kinematic viscosity at the reference state
ρ	kg/m^3	Density
τ	Pa	Shear stress
$\psi = \overline{u'v'} / u' \bullet v'$		Correlation coefficient
$\omega = 2 \pi f$	rad/sec	Angular frequency

Superscripts

+	Wall coordinate
*	Normalized quantity, except where used in friction velocity, u_*

Subscripts

m	Average over cross-section of duct
max	Maximum during one cycle
o	Reference state
w	At the wall

LIST OF FIGURES

DATA LISTINGS

I. SINGLE-WIRE DATA

SPRE Operating Point
Nozzle Inlet Geometry
$(\theta, \tilde{u})$ (θ, u') (θ, u_m)

Ambient Conditions

axial station s/d	T (^{o}C)	P (bar)
0.33	25.68	0.980
16	25.29	0.991
30	25.22	0.989
44	24.06	0.990

Note that the data are tabulated at every 4 degrees of crank position except within ±10 degrees of transition, for which the resolution is every 2 degrees.

II. SUPPLEMENTAL FIGURES

In order to supplement the three-dimensional figures which appear in the main body of the thesis, additional figures generated from the single-wire data have been included after the tabulated data for each of the four axial stations. At each station, plots are provided of the ensemble-averaged velocity, the streamwise rms-velocity fluctuation, and the turbulence intensity.

SPRE s/d = 0.33

θ	r/R = 0.994		r/R = 0.993		r/R = 0.990	
deg.	ū m/sec	u' m/sec	ū m/sec	u' m/sec	ū m/sec	u' m/sec
4	0.0458	0.0071	0.0388	0.0105	0.0340	0.0127
8	0.0536	0.0065	0.0476	0.0088	0.0453	0.0111
12	0.0652	0.0077	0.0616	0.0076	0.0650	0.0118
14	0.0748	0.0094	0.0723	0.0096	0.0826	0.0134
16	0.0845	0.0088	0.0849	0.0105	0.1017	0.0167
18	0.0933	0.0106	0.0962	0.0131	0.1175	0.0184
20	0.1035	0.0139	0.1070	0.0158	0.1329	0.0235
22	0.1077	0.0160	0.1163	0.0193	0.1468	0.0291
24	0.1144	0.0168	0.1274	0.0266	0.1598	0.0329
26	0.1246	0.0211	0.1434	0.0335	0.1827	0.0432
28	0.1484	0.0363	0.1609	0.0372	0.2173	0.0638
30	0.1796	0.0418	0.1851	0.0477	0.2544	0.0779
32	0.1953	0.0427	0.2128	0.0625	0.2956	0.0933
36	0.2452	0.1041	0.2647	0.0916	0.3649	0.1351
40	0.2511	0.0812	0.3002	0.1158	0.4179	0.1507
44	0.3145	0.1099	0.3697	0.1427	0.5204	0.1875
48	0.3483	0.1005	0.4349	0.1446	0.5855	0.1761
52	0.3843	0.1313	0.4700	0.1557	0.6621	0.2103
56	0.3980	0.1170	0.4965	0.1578	0.6913	0.1970
60	0.4376	0.1060	0.5391	0.1576	0.7335	0.1828
64	0.5004	0.1279	0.5925	0.1630	0.7968	0.1878
68	0.5330	0.1542	0.6199	0.1534	0.8054	0.1835
72	0.5970	0.1516	0.6479	0.1390	0.8459	0.1813
76	0.5981	0.1098	0.6719	0.1192	0.8752	0.1673
80	0.5987	0.1004	0.6840	0.0851	0.8909	0.1351
84	0.6010	0.0676	0.6830	0.0577	0.8761	0.0769
88	0.5789	0.0416	0.6649	0.0307	0.8491	0.0479
92	0.5501	0.0252	0.6564	0.0160	0.8316	0.0286
96	0.5372	0.0241	0.6560	0.0059	0.8207	0.0178
100	0.5175	0.0095	0.6560	0.0035	0.8018	0.0078
104	0.5040	0.0085	0.6017	0.0164	0.7732	0.0103
108	0.4790	0.0033	0.5804	0.0048	0.7430	0.0054
112	0.4412	0.0077	0.5312	0.0065	0.7190	0.0057
116	0.4060	0.0070	0.5023	0.0085	0.6766	0.0015
120	0.3866	0.0020	0.4498	0.0074	0.6561	0.0007
124	0.3511	0.0080	0.4071	0.0078	0.5895	0.0111
128	0.3164	0.0080	0.3740	0.0091	0.5238	0.0084
132	0.2714	0.0054	0.3304	0.0036	0.4596	0.0137
136	0.2340	0.0016	0.2696	0.0039	0.3870	0.0023
140	0.1921	0.0020	0.2214	0.0040	0.3010	0.0095
144	0.1540	0.0000	0.1730	0.0031	0.2254	0.0047
148	0.1216	0.0026	0.1291	0.0028	0.1709	0.0045
152	0.0938	0.0036	0.0965	0.0052	0.1205	0.0013

θ	r/R = 0.994		r/R = 0.993		r/R = 0.990	
deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
156	0.0679	0.0026	0.0649	0.0026	0.0700	0.0030
160	0.0455	0.0000	0.0422	0.0026	0.0346	0.0024
164	0.0276	0.0016	0.0217	0.0005	0.0130	0.0003
168	0.0183	0.0023	0.0104	0.0009	0.0042	0.0013
172	0.0133	0.0009	0.0066	0.0001	0.0014	0.0001
176	0.0211	0.0015	0.0121	0.0012	0.0064	0.0008
180	0.0283	0.0009	0.0217	0.0005	0.0130	0.0003

SPRE s/d = 0.33

θ	r/R = 0.988		r/R = 0.984		r/R = 0.981	
deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
4	0.0320	0.0149	0.0423	0.0218	0.0536	0.0304
8	0.0442	0.0139	0.0641	0.0244	0.0858	0.0348
12	0.0699	0.0158	0.1057	0.0270	0.1394	0.0320
14	0.0925	0.0158	0.1433	0.0299	0.1872	0.0366
16	0.1143	0.0211	0.1775	0.0342	0.2334	0.0459
18	0.1346	0.0250	0.2107	0.0372	0.2754	0.0504
20	0.1549	0.0284	0.2447	0.0428	0.3229	0.0632
22	0.1718	0.0404	0.2773	0.0636	0.3653	0.0804
24	0.1904	0.0476	0.3059	0.0789	0.4042	0.0961
26	0.2183	0.0607	0.3507	0.1018	0.4578	0.1126
28	0.2626	0.0816	0.4157	0.1164	0.5185	0.1268
30	0.3105	0.1006	0.4871	0.1351	0.5925	0.1438
32	0.3669	0.1215	0.5607	0.1449	0.6771	0.1756
36	0.4529	0.1551	0.6597	0.1793	0.7750	0.1957
40	0.4928	0.1721	0.7329	0.2225	0.8509	0.2186
44	0.5893	0.2028	0.8490	0.2503	0.9848	0.2862
48	0.6906	0.1961	0.9355	0.2419	1.1050	0.2739
52	0.7504	0.2018	1.0011	0.2572	1.1766	0.2772
56	0.7903	0.2004	1.0701	0.2752	1.2435	0.2858
60	0.8531	0.1911	1.1350	0.2510	1.2968	0.2686
64	0.9001	0.1821	1.2098	0.2494	1.4140	0.2991
68	0.9391	0.2012	1.2400	0.2691	1.4653	0.3162
72	0.9622	0.1989	1.2986	0.2618	1.4999	0.2889
76	1.0067	0.1776	1.3676	0.2694	1.5580	0.2701
80	1.0368	0.1600	1.3817	0.1891	1.5815	0.2124
84	1.0199	0.0873	1.3620	0.1259	1.5686	0.1456
88	0.9927	0.0615	1.3229	0.0810	1.5267	0.0940
92	0.9730	0.0378	1.2925	0.0439	1.4925	0.0489

	r/R = 0.988		r/R = 0.984		r/R = 0.981	
θ deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
96	0.9601	0.0229	1.2794	0.0261	1.4797	0.0263
100	0.9357	0.0125	1.2428	0.0175	1.4556	0.0160
104	0.9166	0.0078	1.2028	0.0069	1.3973	0.0093
108	0.8806	0.0122	1.1767	0.0124	1.3523	0.0069
112	0.8400	0.0085	1.1259	0.0084	1.3129	0.0153
116	0.7902	0.0063	1.0622	0.0040	1.2504	0.0107
120	0.7458	0.0025	1.0082	0.0081	1.1677	0.0041
124	0.6878	0.0073	0.9448	0.0071	1.0927	0.0090
128	0.6530	0.0091	0.8687	0.0070	1.0175	0.0079
132	0.5859	0.0033	0.7954	0.0080	0.9252	0.0076
136	0.4854	0.0128	0.7038	0.0055	0.8338	0.0110
140	0.3785	0.0063	0.6179	0.0042	0.7172	0.0128
144	0.2846	0.0030	0.4717	0.0088	0.5953	0.0096
148	0.2061	0.0099	0.3441	0.0111	0.4506	0.0056
152	0.1395	0.0040	0.2305	0.0071	0.3093	0.0062
156	0.0804	0.0033	0.1317	0.0058	0.1749	0.0074
160	0.0327	0.0030	0.0506	0.0016	0.0705	0.0049
164	0.0082	0.0013	0.0088	0.0006	0.0137	0.0011
168	0.0016	0.0006	0.0001	0.0001	0.0000	0.0000
172	0.0000	0.0000	0.0022	0.0002	0.0041	0.0001
176	0.0019	0.0002	0.0001	0.0001	0.0000	0.0000
180	0.0084	0.0015	0.0086	0.0007	0.0117	0.0008

SPRE s/d = 0.33

	r/R = 0.974		r/R = 0.968		r/R = 0.948	
θ deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
4	0.1161	0.0534	0.1716	0.0830	0.2940	0.1353
8	0.1804	0.0609	0.2774	0.0844	0.4938	0.1285
12	0.2834	0.0629	0.4196	0.0892	0.6983	0.0880
14	0.3701	0.0630	0.5275	0.0876	0.7980	0.0841
16	0.4424	0.0741	0.6188	0.0755	0.9118	0.0878
18	0.5157	0.0816	0.6855	0.0791	1.0048	0.1022
20	0.5816	0.0922	0.7391	0.0796	1.1009	0.1251
22	0.6325	0.0922	0.7879	0.0996	1.1694	0.1413
24	0.6682	0.1049	0.8437	0.1223	1.2577	0.1680
26	0.7241	0.1235	0.9155	0.1531	1.3614	0.1993
28	0.8040	0.1588	1.0036	0.1894	1.4671	0.2347
30	0.8890	0.1785	1.1121	0.2284	1.5998	0.2684
32	0.9858	0.2257	1.2272	0.2704	1.7320	0.3013

	r/R = 0.974		r/R = 0.968		r/R = 0.948	
θ deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
36	1.1154	0.2655	1.3962	0.3193	1.9690	0.3611
40	1.2384	0.3272	1.5175	0.3745	2.1915	0.4221
44	1.3854	0.3739	1.7041	0.4054	2.3792	0.4547
48	1.5686	0.4187	1.8652	0.4388	2.6197	0.4560
52	1.6829	0.4248	2.0333	0.4442	2.8492	0.4453
56	1.7636	0.4207	2.1452	0.4514	2.9400	0.4566
60	1.8286	0.4076	2.2499	0.4518	3.1060	0.4586
64	1.9466	0.4041	2.3708	0.4863	3.2852	0.4592
68	2.0000	0.4324	2.4300	0.4437	3.4210	0.4097
72	2.1032	0.4275	2.5282	0.4380	3.5521	0.4671
76	2.1631	0.3764	2.6479	0.4289	3.6767	0.4325
80	2.2167	0.3059	2.6572	0.3186	3.7516	0.3347
84	2.1763	0.1811	2.6489	0.1956	3.7345	0.2318
88	2.1186	0.1144	2.5982	0.1366	3.6743	0.1620
92	2.0741	0.0684	2.5459	0.0744	3.6151	0.0953
96	2.0519	0.0380	2.5246	0.0450	3.5843	0.0480
100	2.0180	0.0186	2.4691	0.0259	3.5579	0.0190
104	1.9424	0.0135	2.4026	0.0217	3.4661	0.0257
108	1.8980	0.0088	2.3398	0.0012	3.3902	0.0154
112	1.8170	0.0116	2.2580	0.0210	3.2888	0.0000
116	1.7318	0.0114	2.1524	0.0097	3.1533	0.0127
120	1.6515	0.0026	2.0488	0.0179	3.0317	0.0020
124	1.5635	0.0070	1.9355	0.0116	2.9058	0.0079
128	1.4546	0.0116	1.8172	0.0054	2.7336	0.0121
132	1.3361	0.0149	1.6903	0.0157	2.5548	0.0000
136	1.2183	0.0092	1.5457	0.0115	2.3485	0.0137
140	1.0667	0.0124	1.3675	0.0140	2.1189	0.0136
144	0.9028	0.0081	1.1706	0.0085	1.8831	0.0065
148	0.7579	0.0116	0.9944	0.0098	1.6397	0.0143
152	0.6464	0.0020	0.8262	0.0061	1.4033	0.0076
156	0.4054	0.0120	0.6557	0.0070	1.1269	0.0090
160	0.1983	0.0074	0.3804	0.0080	0.8470	0.0083
164	0.0553	0.0030	0.1332	0.0077	0.5494	0.0136
168	0.0022	0.0010	0.0137	0.0017	0.1929	0.0060
172	0.0053	0.0007	0.0038	0.0004	0.0116	0.0014
176	0.0001	0.0000	0.0002	0.0002	0.0077	0.0009
180	0.0277	0.0019	0.0476	0.0038	0.0645	0.0032

SPRE s/d = 0.33

θ	r/R = 0.928		r/R = 0.861		r/R = 0.728	
deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
4	0.3266	0.1696	0.2064	0.1675	0.0785	0.0973
8	0.5746	0.1386	0.5877	0.1775	0.4488	0.1776
12	0.8030	0.1016	0.9220	0.1323	0.8599	0.1336
14	0.9426	0.1024	1.0984	0.1267	1.0638	0.1254
16	1.0669	0.0987	1.2743	0.1249	1.2543	0.1299
18	1.1857	0.1200	1.4324	0.1387	1.4433	0.1316
20	1.2993	0.1349	1.5930	0.1357	1.6206	0.1321
22	1.4153	0.1519	1.7362	0.1352	1.7782	0.1331
24	1.5184	0.1724	1.8556	0.1420	1.9255	0.1233
26	1.6256	0.2134	1.9964	0.1477	2.0803	0.1239
28	1.7611	0.2472	2.1374	0.1661	2.2392	0.1245
30	1.9085	0.2647	2.3062	0.1804	2.4106	0.1138
32	2.0432	0.2823	2.4689	0.1857	2.5652	0.1114
36	2.3248	0.3322	2.7819	0.2007	2.8673	0.1097
40	2.5423	0.3548	3.0143	0.1895	3.1032	0.1094
44	2.7610	0.3824	3.2767	0.1971	3.3486	0.1066
48	3.0323	0.3695	3.5661	0.1517	3.6141	0.0868
52	3.3003	0.3491	3.7929	0.1410	3.8302	0.0754
56	3.4609	0.3552	3.9723	0.1402	3.9991	0.0715
60	3.6057	0.3740	4.1292	0.1368	4.1659	0.0554
64	3.8199	0.3617	4.3411	0.1266	4.3751	0.0567
68	3.9793	0.3285	4.4797	0.0799	4.4914	0.0466
72	4.1334	0.3130	4.5732	0.0683	4.5846	0.0220
76	4.2055	0.2983	4.6705	0.0544	4.6683	0.0201
80	4.3326	0.2363	4.7553	0.0280	4.7447	0.0160
84	4.3651	0.1706	4.8032	0.0331	4.7879	0.0131
88	4.3221	0.1282	4.8065	0.0269	4.7889	0.0103
92	4.2636	0.0809	4.8096	0.0151	4.7906	0.0084
96	4.2407	0.0414	4.8108	0.0134	4.7921	0.0101
100	4.2050	0.0271	4.7825	0.0208	4.7644	0.0213
104	4.1098	0.0125	4.7245	0.0027	4.7086	0.0076
108	4.0408	0.0156	4.6734	0.0181	4.6590	0.0173
112	3.9584	0.0033	4.5935	0.0166	4.5801	0.0153
116	3.8013	0.0171	4.4415	0.0000	4.4261	0.0064
120	3.6538	0.0179	4.2930	0.0190	4.2880	0.0252
124	3.5311	0.0000	4.1414	0.0171	4.1327	0.0194
128	3.3465	0.0207	3.9817	0.0034	3.9691	0.0103
132	3.1395	0.0173	3.7988	0.0033	3.7882	0.0116
136	2.9443	0.0099	3.5836	0.0103	3.5822	0.0163
140	2.6872	0.0244	3.3488	0.0235	3.3680	0.0207
144	2.4137	0.0132	3.0613	0.0114	3.0937	0.0180
148	2.1102	0.0105	2.7826	0.0101	2.8065	0.0135
152	1.8340	0.0189	2.5215	0.0259	2.5389	0.0040

	r/R = 0.928		r/R = 0.861		r/R = 0.728	
θ deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
156	1.5473	0.0100	2.1959	0.0075	2.2465	0.0198
160	1.2105	0.0094	1.8338	0.0104	1.8983	0.0110
164	0.8560	0.0061	1.4620	0.0009	1.5581	0.0030
168	0.5059	0.0100	1.0884	0.0032	1.2130	0.0094
172	0.1423	0.0048	0.7223	0.0018	0.8666	0.0082
176	0.0031	0.0008	0.2353	0.0099	0.4639	0.0078
180	0.0228	0.0019	0.0020	0.0008	0.0295	0.0027

SPRE s/d = 0.33

	r/R = 0.594		r/R = 0.461		r/R = 0.328	
θ deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
4	0.0536	0.0773	0.0360	0.0568	0.0199	0.0387
8	0.3680	0.1685	0.2841	0.1526	0.2486	0.1416
12	0.8018	0.1317	0.7402	0.1255	0.6919	0.1225
14	1.0091	0.1256	0.9506	0.1249	0.8937	0.1031
16	1.2106	0.1319	1.1448	0.1212	1.1083	0.1165
18	1.4118	0.1296	1.3646	0.1337	1.3288	0.1394
20	1.5987	0.1307	1.5587	0.1342	1.5526	0.1432
22	1.7724	0.1376	1.7315	0.1339	1.7434	0.1352
24	1.9308	0.1363	1.9090	0.1348	1.9112	0.1382
26	2.0906	0.1272	2.0599	0.1388	2.0640	0.1400
28	2.2451	0.1311	2.2385	0.1387	2.2378	0.1461
30	2.4241	0.1295	2.4150	0.1164	2.4201	0.1424
32	2.5848	0.1180	2.5801	0.1210	2.5780	0.1347
36	2.8912	0.1164	2.8758	0.1239	2.8775	0.1310
40	3.1227	0.1086	3.0961	0.1212	3.0999	0.1165
44	3.3684	0.1032	3.3487	0.1098	3.3514	0.1199
48	3.6343	0.0892	3.5965	0.0922	3.5977	0.0944
52	3.8378	0.0803	3.7947	0.0865	3.7998	0.0930
56	4.0079	0.0706	3.9623	0.0719	3.9623	0.0874
60	4.1622	0.0658	4.1259	0.0697	4.1178	0.0667
64	4.3672	0.0580	4.3343	0.0607	4.3144	0.0662
68	4.4824	0.0432	4.4454	0.0527	4.4337	0.0472
72	4.5884	0.0246	4.5372	0.0240	4.5348	0.0337
76	4.6681	0.0207	4.6246	0.0208	4.6137	0.0304
80	4.7408	0.0207	4.6954	0.0124	4.6823	0.0175
84	4.7817	0.0208	4.7342	0.0148	4.7182	0.0131
88	4.7830	0.0205	4.7340	0.0136	4.7183	0.0129
92	4.7917	0.0157	4.7393	0.0100	4.7226	0.0172

	r/R = 0.594		r/R = 0.461		r/R = 0.328	
θ deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
96	4.7951	0.0116	4.7413	0.0131	4.7267	0.0196
100	4.7581	0.0067	4.7094	0.0194	4.7114	0.0119
104	4.7155	0.0032	4.6562	0.0037	4.6653	0.0171
108	4.6379	0.0215	4.6053	0.0187	4.5631	0.0189
112	4.5643	0.0185	4.5281	0.0151	4.5033	0.0182
116	4.4332	0.0000	4.3768	0.0053	4.3927	0.0040
120	4.2783	0.0091	4.2358	0.0237	4.2283	0.0172
124	4.1266	0.0109	4.0826	0.0185	4.0781	0.0155
128	3.9740	0.0029	3.9238	0.0106	3.9364	0.0041
132	3.7916	0.0033	3.7440	0.0107	3.7535	0.0091
136	3.5788	0.0092	3.5383	0.0157	3.5339	0.0167
140	3.3559	0.0217	3.3228	0.0207	3.3021	0.0233
144	3.0817	0.0174	3.0532	0.0182	3.0333	0.0172
148	2.8046	0.0086	2.7733	0.0151	2.7696	0.0131
152	2.5438	0.0000	2.5056	0.0000	2.5167	0.0028
156	2.2304	0.0164	2.2211	0.0196	2.1986	0.0121
160	1.8978	0.0050	1.8746	0.0117	1.8753	0.0045
164	1.5613	0.0021	1.5343	0.0000	1.5418	0.0038
168	1.2106	0.0058	1.1973	0.0089	1.1952	0.0067
172	0.8641	0.0053	0.8620	0.0094	0.8552	0.0080
176	0.4648	0.0087	0.4546	0.0081	0.4560	0.0073
180	0.0341	0.0038	0.0354	0.0030	0.0362	0.0032

SPRE s/d = 0.33

	r/R = 0.194		r/R = 0.006	
θ deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
4	0.0205	0.0365	0.0241	0.0486
8	0.2157	0.1299	0.2084	0.1402
12	0.6604	0.1171	0.6475	0.1280
14	0.8715	0.1041	0.8665	0.1103
16	1.0866	0.1120	1.0795	0.1173
18	1.3080	0.1286	1.3149	0.1249
20	1.5347	0.1388	1.5472	0.1336
22	1.7324	0.1372	1.7456	0.1431
24	1.9053	0.1491	1.9121	0.1418
26	2.0707	0.1399	2.0792	0.1377
28	2.2340	0.1530	2.2517	0.1468
30	2.4150	0.1435	2.4257	0.1474
32	2.5871	0.1345	2.5843	0.1290
36	2.8926	0.1234	2.8915	0.1221

	r/R = 0.194		r/R = 0.006	
θ deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
40	3.1142	0.1231	3.1064	0.1201
44	3.3570	0.1179	3.3511	0.1178
48	3.5990	0.1116	3.6026	0.1034
52	3.8103	0.0989	3.8128	0.0970
56	3.9553	0.0886	3.9673	0.0873
60	4.1152	0.0754	4.1305	0.0802
64	4.3148	0.0662	4.3202	0.0659
68	4.4303	0.0483	4.4401	0.0605
72	4.5268	0.0379	4.5225	0.0372
76	4.6109	0.0292	4.6138	0.0213
80	4.6819	0.0173	4.6817	0.0184
84	4.7153	0.0085	4.7248	0.0198
88	4.7150	0.0072	4.7225	0.0190
92	4.7192	0.0140	4.7309	0.0209
96	4.7260	0.0192	4.7356	0.0209
100	4.7120	0.0088	4.7083	0.0125
104	4.6635	0.0176	4.6398	0.0175
108	4.5630	0.0188	4.5897	0.0052
112	4.5045	0.0167	4.5097	0.0018
116	4.3923	0.0000	4.3613	0.0171
120	4.2272	0.0169	4.2501	0.0279
124	4.0786	0.0144	4.0772	0.0155
128	3.9361	0.0000	3.9130	0.0210
132	3.7530	0.0083	3.7358	0.0181
136	3.5336	0.0163	3.5381	0.0197
140	3.3027	0.0220	3.3222	0.0192
144	3.0339	0.0163	3.0489	0.0079
148	2.7693	0.0130	2.7782	0.0198
152	2.5164	0.0000	2.4935	0.0101
156	2.1977	0.0108	2.2200	0.0062
160	1.8747	0.0026	1.8731	0.0090
164	1.5412	0.0042	1.5266	0.0083
168	1.1943	0.0061	1.1944	0.0025
172	0.8541	0.0073	0.8655	0.0017
176	0.4552	0.0068	0.4636	0.0090
180	0.0371	0.0027	0.0371	0.0030

SPRE s/d = 0.33

θ deg.	u_m m/sec	θ deg.	u_m m/sec	θ deg.	u_m m/sec
2	0.0740	90	4.8078	178	0.1559
4	0.1416	92	4.7992	180	0.0472
6	0.2622	94	4.7978		
8	0.4277	96	4.7884		
10	0.6108	98	4.7725		
12	0.7978	100	4.7460		
14	0.9845	102	4.7172		
16	1.1653	104	4.6798		
18	1.3450	106	4.6412		
20	1.5180	108	4.5997		
22	1.6693	110	4.5330		
24	1.8101	112	4.4992		
26	1.9557	114	4.4107		
28	2.1193	116	4.3390		
30	2.3029	118	4.2788		
32	2.4641	120	4.1857		
34	2.6323	122	4.1142		
36	2.7818	124	4.0183		
38	2.9018	126	3.9408		
40	3.0013	128	3.8419		
42	3.1438	130	3.7323		
44	3.2800	132	3.6355		
46	3.4149	134	3.5312		
48	3.5546	136	3.4095		
50	3.7027	138	3.3132		
52	3.7859	140	3.1635		
54	3.8752	142	3.0286		
56	3.9519	144	2.8760		
58	4.0422	146	2.7438		
60	4.1301	148	2.5885		
62	4.2519	150	2.4567		
64	4.3690	152	2.3195		
66	4.4419	154	2.1730		
68	4.5087	156	2.0135		
70	4.6048	158	1.8552		
72	4.6516	160	1.6741		
74	4.6792	162	1.5017		
76	4.7371	164	1.3370		
78	4.7829	166	1.1752		
80	4.8121	168	1.0047		
82	4.8371	170	0.8403		
84	4.8478	172	0.6877		
86	4.8395	174	0.5349		
88	4.8248	176	0.3457		

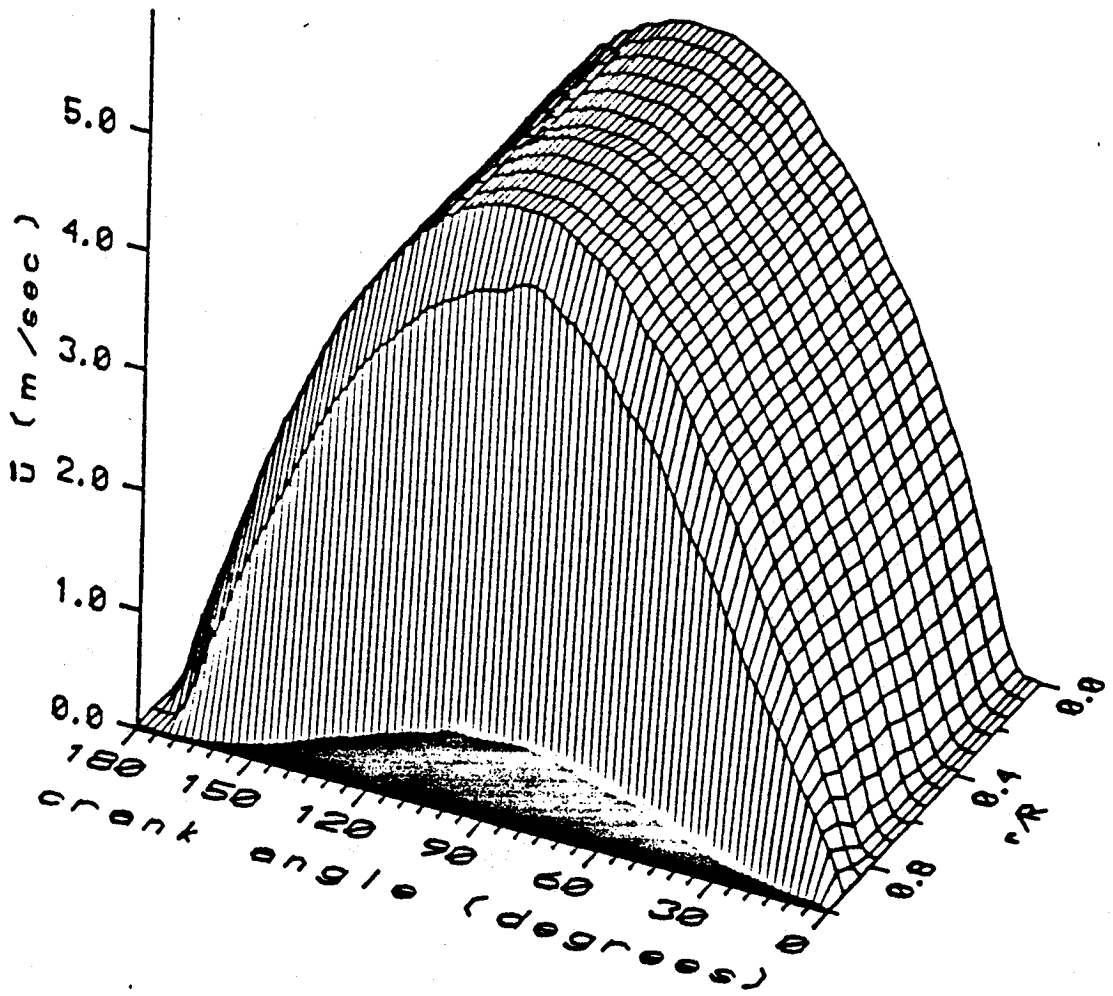


Figure 1: Ensemble-averaged velocity at s/d = 0.33

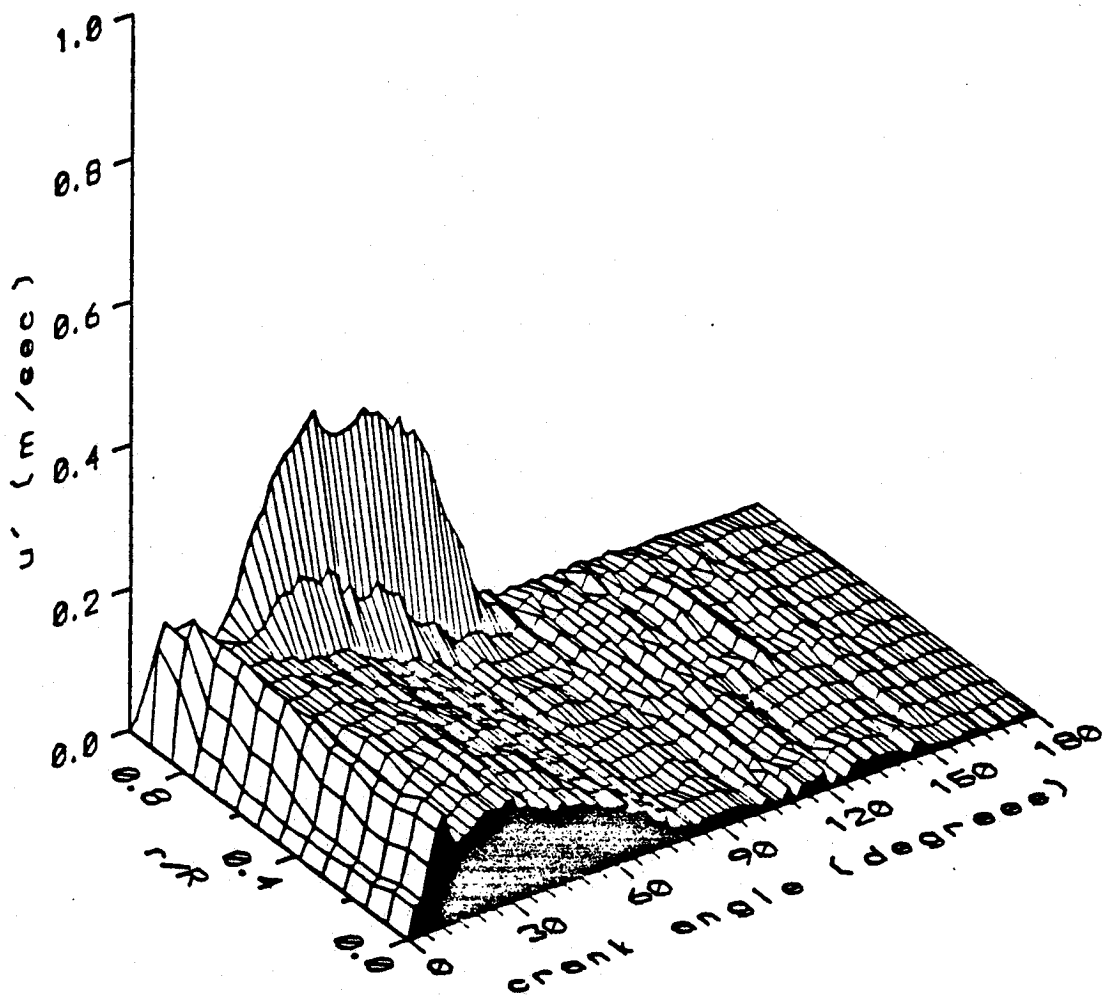


Figure 2: Streamwise velocity fluctuation at s/d = 0.33

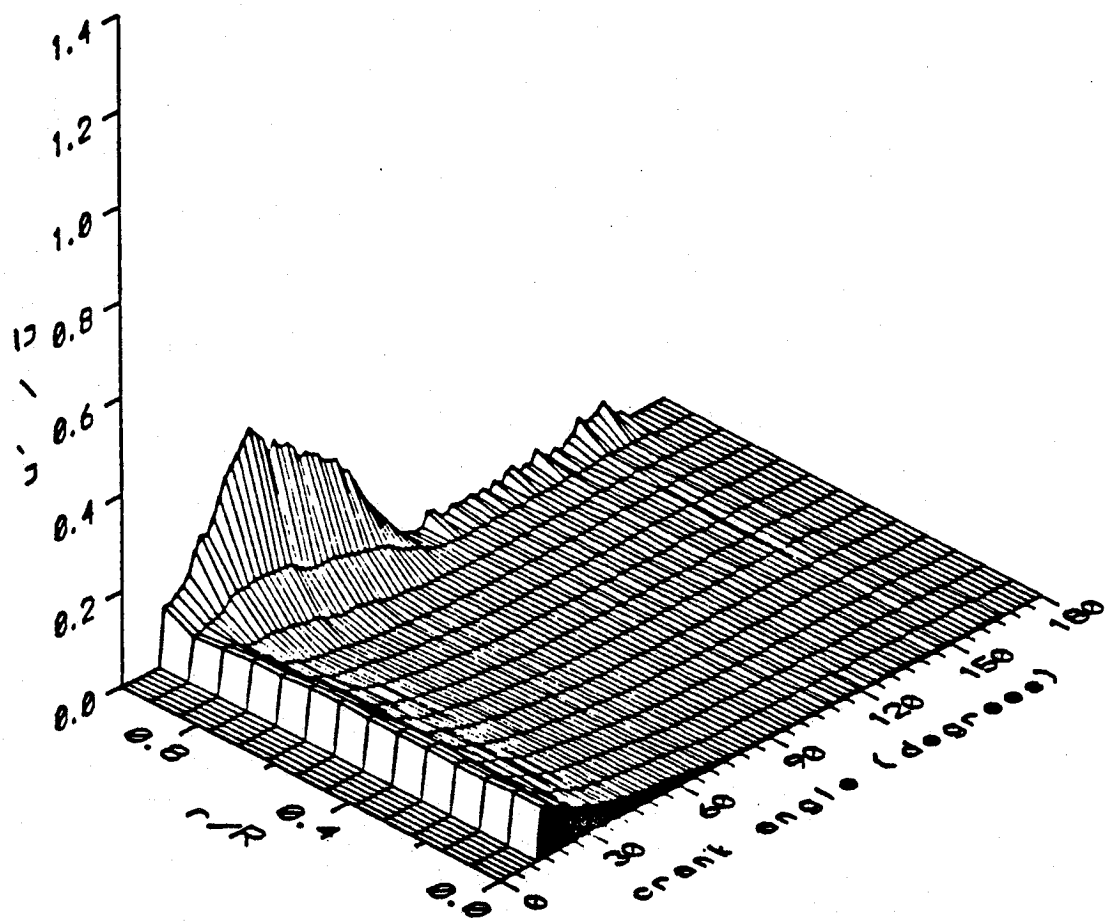


Figure 3: Streamwise turbulence intensity, u' / ū , at s/d = 0.33

SPRE s/d = 16

θ deg.	r/R = 0.995 ũ m/sec	r/R = 0.995 u' m/sec	r/R = 0.992 ũ m/sec	r/R = 0.992 u' m/sec	r/R = 0.983 ũ m/sec	r/R = 0.983 u' m/sec
4	0.0499	0.0091	0.0392	0.0119	0.0644	0.0282
8	0.0573	0.0087	0.0495	0.0100	0.0974	0.0301
12	0.0727	0.0080	0.0690	0.0112	0.1587	0.0320
16	0.0973	0.0072	0.0999	0.0100	0.2471	0.0328
20	0.1131	0.0098	0.1280	0.0131	0.3371	0.0380
24	0.1350	0.0298	0.1624	0.0561	0.4368	0.1359
28	0.2114	0.0510	0.2794	0.0806	0.6894	0.1521
32	0.2655	0.0707	0.3674	0.1130	0.8406	0.1750
36	0.2394	0.0413	0.3264	0.0671	0.7925	0.1097
40	0.2293	0.0224	0.3019	0.0383	0.7531	0.0607
44	0.2425	0.0162	0.3263	0.0296	0.7823	0.0461
48	0.2605	0.0181	0.3537	0.0261	0.8329	0.0465
52	0.2655	0.0197	0.3611	0.0272	0.8460	0.0480
54	0.2662	0.0194	0.3627	0.0252	0.8515	0.0472
56	0.2694	0.0210	0.3659	0.0251	0.8547	0.0479
58	0.2760	0.0226	0.3745	0.0271	0.8672	0.0493
60	0.2877	0.0271	0.3910	0.0354	0.8930	0.0614
62	0.3030	0.0366	0.4123	0.0562	0.9260	0.0939
64	0.3237	0.0588	0.4514	0.0950	0.9688	0.1438
66	0.3506	0.1015	0.4897	0.1306	1.0288	0.2312
68	0.3903	0.1464	0.5284	0.1946	1.1187	0.3508
70	0.4289	0.1799	0.5820	0.2273	1.2125	0.4251
72	0.4817	0.2317	0.6453	0.2757	1.3152	0.5137
76	0.5640	0.2439	0.7760	0.3438	1.4974	0.5860
80	0.6581	0.3043	0.8577	0.3539	1.6698	0.6206
84	0.6855	0.2908	0.8772	0.3578	1.7056	0.6200
88	0.6994	0.2917	0.9276	0.3893	1.7444	0.6151
92	0.6983	0.3068	0.9031	0.3658	1.8339	0.6365
96	0.6762	0.2712	0.9327	0.3724	1.7243	0.6100
100	0.6553	0.3134	0.8509	0.3702	1.6074	0.5988
104	0.5226	0.2693	0.7245	0.3182	1.4033	0.5402
108	0.3883	0.1664	0.5261	0.2304	1.0754	0.3322
112	0.2905	0.1119	0.3969	0.1545	0.8844	0.2284
116	0.2348	0.0489	0.3159	0.0829	0.7572	0.1671
120	0.1918	0.0335	0.2461	0.0571	0.6437	0.1164
124	0.1596	0.0261	0.1963	0.0401	0.5362	0.1057
128	0.1391	0.0211	0.1626	0.0300	0.4469	0.0911
132	0.1215	0.0162	0.1386	0.0226	0.3789	0.0688
136	0.1016	0.0087	0.1083	0.0137	0.2905	0.0464
140	0.0812	0.0084	0.0792	0.0096	0.2041	0.0276
144	0.0655	0.0055	0.0572	0.0056	0.1415	0.0194
148	0.0518	0.0036	0.0412	0.0046	0.0850	0.0136
152	0.0385	0.0037	0.0243	0.0031	0.0384	0.0073

θ	r/R = 0.995		r/R = 0.992		r/R = 0.983	
deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
156	0.0251	0.0025	0.0117	0.0025	0.0066	0.0028
160	0.0178	0.0005	0.0045	0.0008	0.0004	0.0003
164	0.0177	0.0006	0.0041	0.0008	0.0008	0.0004
168	0.0229	0.0029	0.0094	0.0022	0.0026	0.0021
172	0.0320	0.0045	0.0166	0.0033	0.0119	0.0058
176	0.0387	0.0049	0.0239	0.0052	0.0311	0.0115
180	0.0463	0.0063	0.0342	0.0075	0.0625	0.0228

SPRE s/d = 16

θ	r/R = 0.959		r/R = 0.939		r/R = 0.872	
deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
4	0.2712	0.1173	0.3455	0.1541	0.2734	0.1933
8	0.4204	0.1100	0.5951	0.1155	0.6513	0.1579
12	0.6245	0.0839	0.8039	0.0867	0.9706	0.1198
16	0.8123	0.0716	1.0510	0.0864	1.3223	0.1023
20	0.9821	0.0675	1.2756	0.0944	1.6846	0.0985
24	1.1899	0.2423	1.5357	0.2714	2.1845	0.2711
28	1.6245	0.2550	2.0798	0.2685	2.5694	0.1983
32	1.8034	0.2973	2.1906	0.2888	2.6259	0.2316
36	1.7522	0.1943	2.2501	0.2117	3.0017	0.1335
40	1.7236	0.1329	2.2749	0.1593	3.1958	0.1268
44	1.7790	0.1141	2.3620	0.1391	3.3970	0.1304
48	1.8756	0.1109	2.4760	0.1363	3.6153	0.1507
52	1.9279	0.1105	2.5506	0.1368	3.7920	0.1679
54	1.9458	0.1114	2.5754	0.1393	3.8703	0.1646
56	1.9567	0.1066	2.6053	0.1413	3.9356	0.1704
58	1.9840	0.1185	2.6365	0.1542	4.0149	0.2123
60	2.0253	0.1534	2.7007	0.2056	4.1113	0.2938
62	2.0782	0.2377	2.8200	0.3185	4.2186	0.3812
64	2.1664	0.3377	2.9307	0.4399	4.3262	0.4421
66	2.2941	0.4636	3.0790	0.5486	4.3762	0.5166
68	2.4729	0.5986	3.1904	0.6870	4.3871	0.6006
70	2.5574	0.6174	3.3052	0.6851	4.4580	0.6324
72	2.6912	0.6982	3.4395	0.7268	4.5265	0.5600
76	2.9505	0.7765	3.5936	0.7247	4.5987	0.6451
80	3.1147	0.7300	3.7722	0.7423	4.6843	0.6279
84	3.1667	0.7535	3.8012	0.7265	4.6791	0.6911
88	3.2034	0.7945	3.8137	0.7587	4.7590	0.6584
92	3.2329	0.7436	3.9123	0.7419	4.7822	0.6262

	r/R = 0.959		r/R = 0.939		r/R = 0.872	
θ deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
96	3.2303	0.7264	3.8506	0.7579	4.6250	0.6642
100	3.1196	0.7759	3.6977	0.7479	4.6618	0.6683
104	2.7603	0.6706	3.4730	0.6850	4.6415	0.6343
108	2.3775	0.5449	3.1118	0.5649	4.4928	0.5705
112	2.0545	0.4237	2.7702	0.4636	4.2916	0.5018
116	1.8296	0.2960	2.5221	0.3778	4.0751	0.4266
120	1.6246	0.2295	2.2985	0.3056	3.8111	0.3749
124	1.4503	0.1929	2.0852	0.2577	3.5616	0.3009
128	1.3206	0.1604	1.9146	0.2037	3.3640	0.2250
132	1.2210	0.1217	1.7815	0.1516	3.1915	0.1532
136	1.0912	0.0828	1.6190	0.1060	2.9934	0.1072
140	0.9487	0.0591	1.4425	0.0758	2.7773	0.0888
144	0.8158	0.0419	1.2759	0.0598	2.5430	0.0693
148	0.6797	0.0330	1.0929	0.0468	2.2817	0.0606
152	0.4918	0.0426	0.8939	0.0438	1.9845	0.0537
156	0.2399	0.0363	0.6558	0.0369	1.6377	0.0497
160	0.0451	0.0178	0.3037	0.0498	1.2701	0.0532
164	0.0019	0.0018	0.0464	0.0243	0.8946	0.0545
168	0.0012	0.0017	0.0015	0.0015	0.5346	0.0796
172	0.0365	0.0194	0.0095	0.0114	0.1501	0.0561
176	0.1415	0.0389	0.1273	0.0476	0.0035	0.0094
180	0.3169	0.0708	0.3423	0.0843	0.0901	0.0788

SPRE s/d = 16

	r/R = 0.739		r/R = 0.606		r/R = 0.472	
θ deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
4	0.1116	0.1144	0.0688	0.0899	0.0424	0.0642
8	0.5178	0.1790	0.4176	0.1660	0.3498	0.1605
12	0.9392	0.1270	0.8605	0.1251	0.8083	0.1196
16	1.3461	0.1148	1.2908	0.1180	1.2407	0.1131
20	1.7467	0.1074	1.7132	0.1107	1.6615	0.1046
24	2.3806	0.2035	2.2267	0.1957	2.0631	0.1701
28	2.4453	0.1780	2.3378	0.1499	2.3003	0.1482
32	2.7765	0.1871	2.7691	0.1911	2.7470	0.1915
36	3.2040	0.1207	3.1975	0.1496	3.0965	0.1884
40	3.4855	0.0991	3.5000	0.1159	3.4625	0.1780
44	3.7694	0.0847	3.8068	0.1054	3.7733	0.1317
48	4.0854	0.0777	4.1305	0.0800	4.1233	0.1000
52	4.3565	0.0819	4.4165	0.0737	4.4060	0.1037

	r/R = 0.739		r/R = 0.606		r/R = 0.472	
θ deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
54	4.4564	0.0877	4.5282	0.0728	4.5145	0.0945
56	4.5460	0.0970	4.6201	0.0777	4.6072	0.0923
58	4.6441	0.1328	4.7118	0.1019	4.6910	0.1135
60	4.7242	0.1744	4.8097	0.1256	4.7953	0.1258
62	4.8175	0.2135	4.9174	0.1448	4.9115	0.1457
64	4.8871	0.2730	4.9926	0.2011	5.0080	0.1627
66	4.9709	0.3124	5.0740	0.2234	5.1085	0.1861
68	4.9945	0.3369	5.1361	0.2507	5.1514	0.2099
70	5.0282	0.3888	5.1715	0.2626	5.1933	0.2215
72	5.0606	0.4027	5.2006	0.2612	5.2199	0.2290
76	5.0799	0.4130	5.2639	0.2873	5.3034	0.2395
80	5.1701	0.4340	5.3242	0.3142	5.3655	0.2616
84	5.2281	0.4285	5.3820	0.3112	5.4701	0.2372
88	5.1939	0.4465	5.4097	0.3050	5.4590	0.2634
92	5.2265	0.4273	5.3949	0.3425	5.4834	0.2467
96	5.1845	0.4616	5.4570	0.3007	5.5353	0.2253
100	5.2515	0.4279	5.4757	0.2504	5.5468	0.1804
104	5.2698	0.3668	5.4608	0.1933	5.5070	0.1367
108	5.2460	0.3281	5.4278	0.1535	5.4607	0.0789
112	5.2041	0.2501	5.3889	0.0883	5.3945	0.0475
116	5.1239	0.1712	5.3437	0.0527	5.3344	0.0488
120	4.9686	0.1252	5.2029	0.0366	5.1900	0.0369
124	4.7608	0.0848	5.0021	0.0217	4.9856	0.0264
128	4.5829	0.0648	4.8185	0.0298	4.8060	0.0309
132	4.3841	0.0358	4.6262	0.0260	4.6182	0.0276
136	4.1427	0.0399	4.3969	0.0141	4.3851	0.0194
140	3.9065	0.0383	4.1325	0.0276	4.1333	0.0319
144	3.5968	0.0275	3.8456	0.0341	3.8473	0.0491
148	3.2763	0.0334	3.5179	0.0316	3.5153	0.0484
152	2.9292	0.0332	3.1947	0.0241	3.2047	0.0252
156	2.5410	0.0276	2.8543	0.0248	2.8647	0.0228
160	2.1576	0.0305	2.5041	0.0268	2.5195	0.0222
164	1.7451	0.0367	2.0838	0.0237	2.1327	0.0289
168	1.3208	0.0386	1.6599	0.0264	1.7171	0.0274
172	0.9026	0.0487	1.2445	0.0326	1.3328	0.0310
176	0.4449	0.0813	0.8120	0.0314	0.9081	0.0344
180	0.0329	0.0429	0.2748	0.0539	0.4105	0.0561

SPRE s/d = 16

θ	r/R = 0.339		r/R = 0.206		r/R = 0.006	
deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
4	0.0307	0.0502	0.0197	0.0362	0.0244	0.0490
8	0.3098	0.1530	0.2601	0.1361	0.2618	0.1471
12	0.7632	0.1158	0.7315	0.1042	0.7339	0.1175
16	1.1881	0.1107	1.1558	0.1020	1.1586	0.0995
20	1.6083	0.1012	1.5760	0.0923	1.5912	0.0943
24	1.9532	0.1289	1.9170	0.1316	1.9189	0.1113
28	2.3215	0.1611	2.3364	0.1734	2.3739	0.1911
32	2.7292	0.1850	2.7554	0.1960	2.8076	0.1782
36	3.0128	0.1971	2.9521	0.1996	3.0080	0.1975
40	3.3378	0.2212	3.2128	0.2379	3.1900	0.2268
44	3.6811	0.1881	3.5317	0.2513	3.3847	0.2498
48	4.0856	0.1455	3.9364	0.2420	3.6980	0.2901
52	4.3624	0.1639	4.2399	0.2398	3.9511	0.3327
54	4.4901	0.1447	4.3735	0.2277	4.0778	0.3288
56	4.5750	0.1462	4.4765	0.2303	4.2018	0.3257
58	4.6726	0.1539	4.5851	0.2130	4.3386	0.3350
60	4.7762	0.1522	4.6792	0.2556	4.4371	0.3542
62	4.8774	0.1760	4.8086	0.2448	4.5971	0.3527
64	4.9949	0.2083	4.8853	0.2689	4.7304	0.3672
66	5.0950	0.2094	4.9780	0.2957	4.7498	0.3994
68	5.1548	0.2273	5.0118	0.2985	4.8220	0.4127
70	5.1811	0.2363	5.0851	0.2947	4.8815	0.3915
72	5.2094	0.2370	5.1100	0.3282	4.9306	0.3944
76	5.2870	0.2414	5.2255	0.2838	5.0325	0.4370
80	5.3754	0.2391	5.3141	0.3086	5.1720	0.4016
84	5.4589	0.2430	5.3982	0.2922	5.1702	0.4263
88	5.4662	0.2348	5.4191	0.2753	5.2517	0.3897
92	5.5146	0.2277	5.4520	0.2776	5.3063	0.3950
96	5.5687	0.1820	5.5147	0.2310	5.3318	0.3963
100	5.5682	0.1591	5.5389	0.2010	5.3816	0.3688
104	5.5272	0.1149	5.5005	0.1440	5.3401	0.3290
108	5.4600	0.0829	5.4240	0.1623	5.2988	0.3291
112	5.4028	0.0614	5.3678	0.1256	5.2242	0.3244
116	5.3240	0.0676	5.2957	0.1087	5.1611	0.2849
120	5.1796	0.0665	5.1540	0.0967	5.0414	0.2825
124	4.9825	0.0468	4.9729	0.0891	4.8835	0.2502
128	4.8027	0.0478	4.7880	0.0693	4.7347	0.1963
132	4.6067	0.0417	4.6006	0.0591	4.5606	0.1720
136	4.3889	0.0242	4.3887	0.0499	4.3532	0.1614
140	4.1304	0.0434	4.1177	0.0685	4.0798	0.1743
144	3.8434	0.0700	3.8128	0.0997	3.7515	0.1674
148	3.5009	0.0843	3.4782	0.1174	3.4945	0.1204
152	3.2049	0.0313	3.1927	0.0491	3.2137	0.0892

	r/R = 0.339		r/R = 0.206		r/R = 0.006	
θ deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
156	2.8672	0.0217	2.8655	0.0271	2.9101	0.0433
160	2.5216	0.0197	2.5225	0.0238	2.5576	0.0399
164	2.1343	0.0293	2.1342	0.0313	2.1819	0.0447
168	1.7197	0.0274	1.7187	0.0289	1.7602	0.0402
172	1.3427	0.0323	1.3438	0.0343	1.3626	0.0413
176	0.9182	0.0363	0.9194	0.0399	0.9189	0.0694
180	0.4282	0.0617	0.4353	0.0761	0.4043	0.1078

SPRE s/d = 16

θ deg.	u_m m/sec	θ deg.	u_m m/sec	θ deg.	u_m m/sec
2	0.0542	90	4.8057	178	0.3664
4	0.1313	92	4.8399	180	0.2327
6	0.2641	94	4.8355		
8	0.4416	96	4.8221		
10	0.6280	98	4.8369		
12	0.8126	100	4.8264		
14	0.9908	102	4.8016		
16	1.1783	104	4.7701		
18	1.3633	106	4.7267		
20	1.5385	108	4.6585		
22	1.7365	110	4.5971		
24	1.9722	112	4.5422		
26	2.1000	114	4.4911		
28	2.2359	116	4.4274		
30	2.3868	118	4.3448		
32	2.5283	120	4.2560		
34	2.6990	122	4.1484		
36	2.8306	124	4.0545		
38	2.9497	126	3.9647		
40	3.0612	128	3.8813		
42	3.1796	130	3.7915		
44	3.3041	132	3.7088		
46	3.4475	134	3.6141		
48	3.5810	136	3.5036		
50	3.7007	138	3.4002		
52	3.8023	140	3.2785		
54	3.8915	142	3.1577		
56	3.9656	144	3.0246		
58	4.0454	146	2.8944		
60	4.1311	148	2.7445		
62	4.2324	150	2.6067		
64	4.3206	152	2.4602		
66	4.4063	154	2.2945		
68	4.4569	156	2.1410		
70	4.5111	158	1.9767		
72	4.5628	160	1.8088		
74	4.6023	162	1.6373		
76	4.6466	164	1.4604		
78	4.6947	166	1.2917		
80	4.7405	168	1.1223		
82	4.7683	170	0.9546		
84	4.7924	172	0.7988		
86	4.8083	174	0.6487		
88	4.8099	176	0.5042		

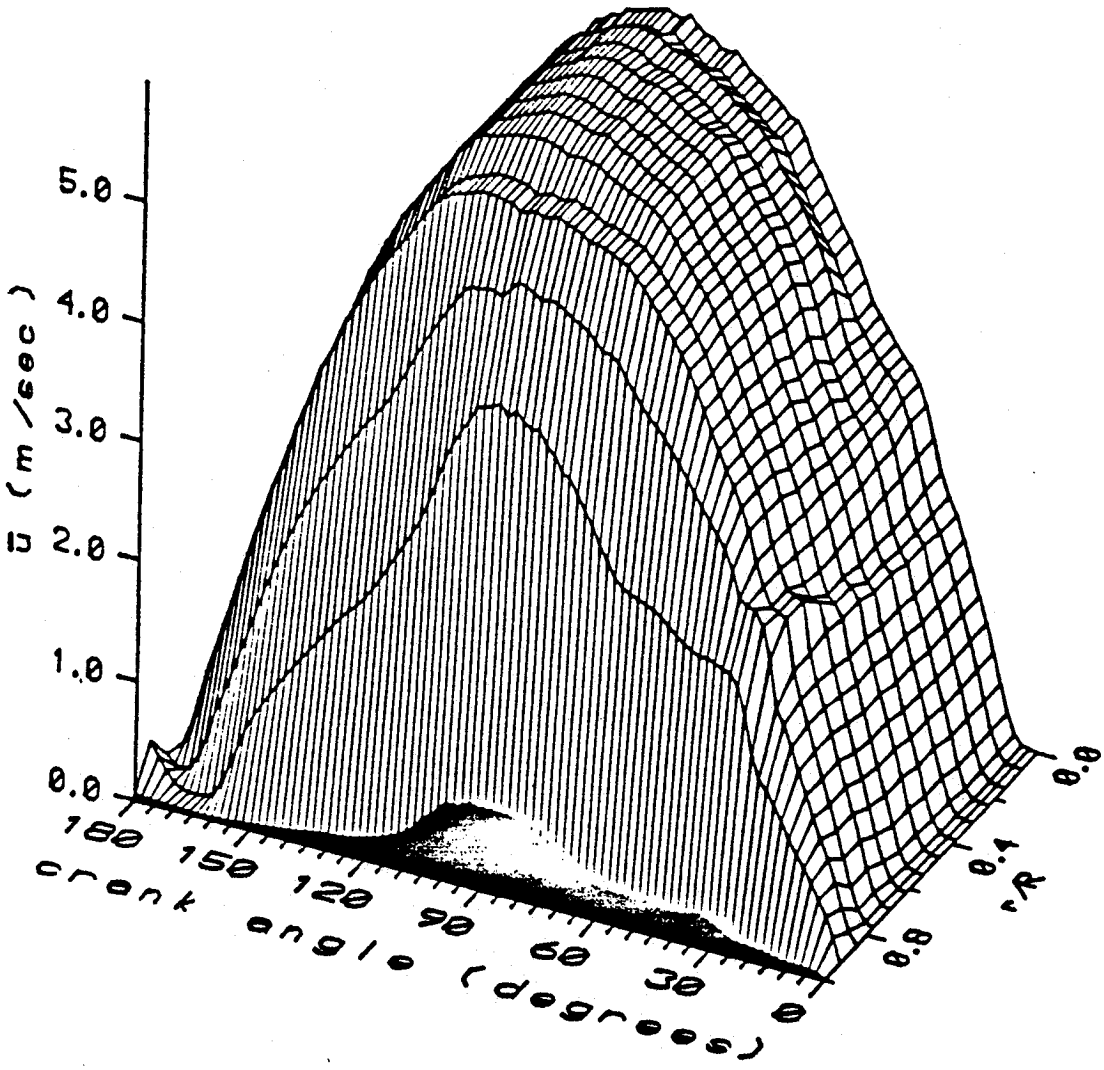


Figure 4: Ensemble-averaged velocity at s/d = 16

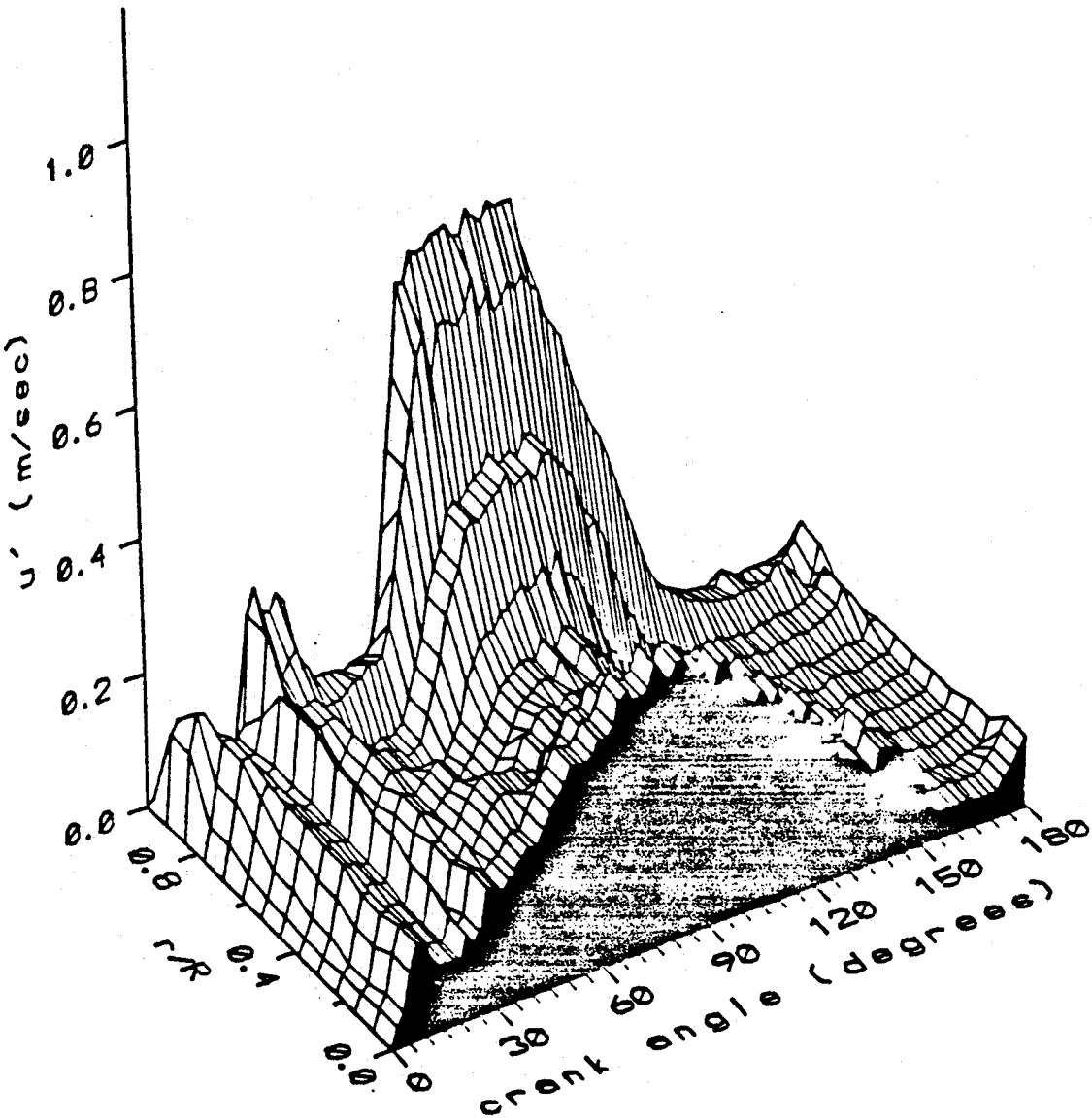


Figure 5: Streamwise velocity fluctuation at s/d = 16

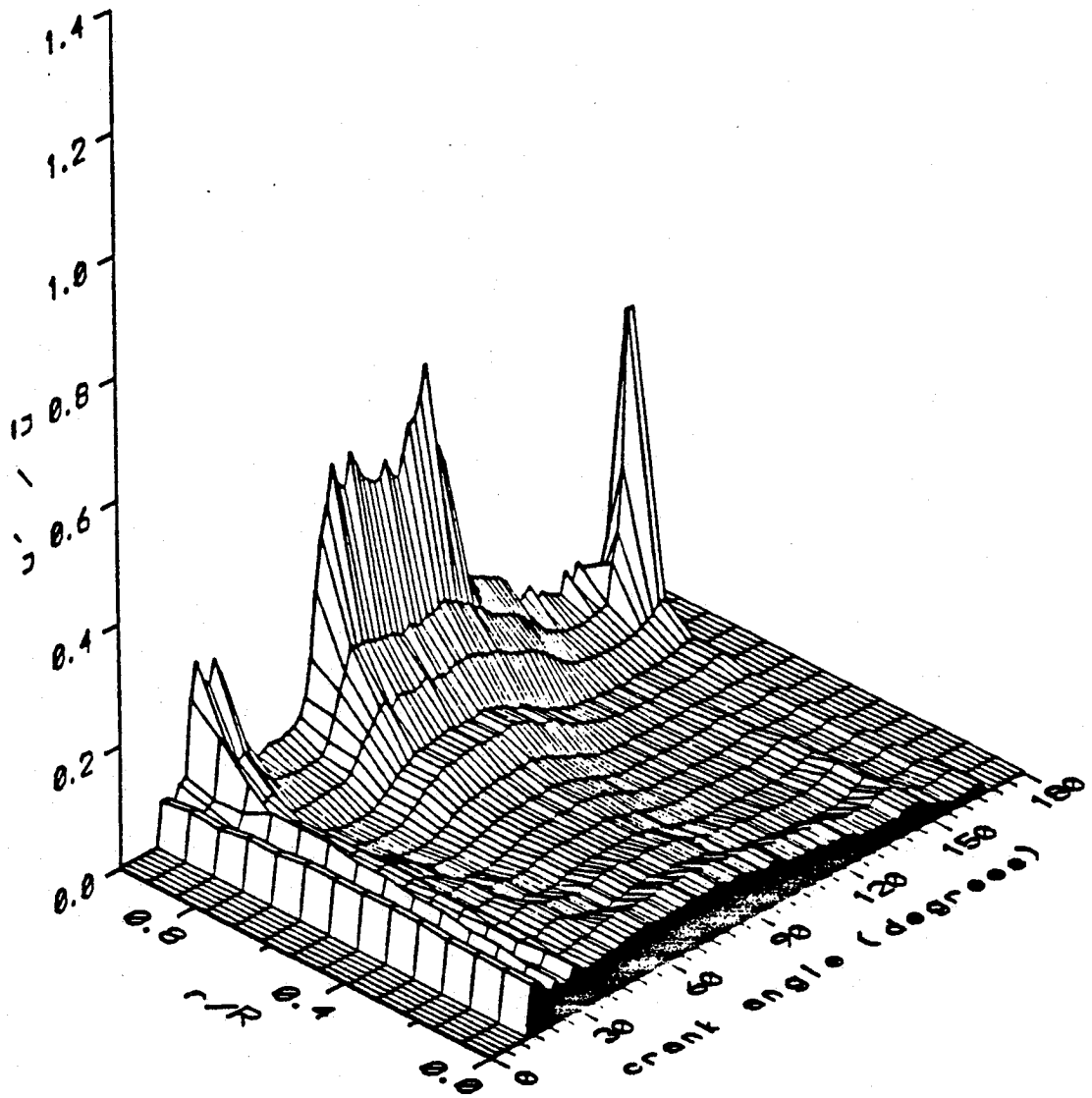


Figure 6: Streamwise turbulence intensity, u' / ū , at s/d = 16

Note: The peaks appear lower than the actual data due to smoothing by the plotting package. The peak turbulence intensity is 1.02, at 158° for r/R = 0.983.

SPRE s/d = 30

θ	r/R = 0.997		r/R = 0.995		r/R = 0.991	
deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
4	0.0483	0.0107	0.0408	0.0120	0.0285	0.0159
8	0.0577	0.0097	0.0504	0.0102	0.0437	0.0175
12	0.0724	0.0093	0.0651	0.0094	0.0675	0.0169
16	0.0941	0.0093	0.0901	0.0108	0.1078	0.0209
20	0.1089	0.0084	0.1097	0.0113	0.1461	0.0210
24	0.1235	0.0115	0.1267	0.0120	0.1750	0.0202
28	0.1406	0.0107	0.1474	0.0138	0.2092	0.0230
32	0.1647	0.0102	0.1713	0.0108	0.2589	0.0272
36	0.1767	0.0099	0.1901	0.0144	0.3005	0.0292
40	0.1833	0.0123	0.1997	0.0153	0.3204	0.0281
44	0.1978	0.0130	0.2174	0.0178	0.3496	0.0299
48	0.2176	0.0152	0.2396	0.0155	0.3941	0.0349
52	0.2261	0.0143	0.2475	0.0156	0.4091	0.0352
56	0.2259	0.0146	0.2471	0.0167	0.4083	0.0359
60	0.2315	0.0140	0.2550	0.0184	0.4246	0.0353
64	0.2384	0.0117	0.2651	0.0199	0.4431	0.0366
68	0.2383	0.0119	0.2660	0.0199	0.4433	0.0359
72	0.2291	0.0136	0.2541	0.0183	0.4210	0.0344
76	0.2228	0.0147	0.2450	0.0153	0.4029	0.0333
78	0.2230	0.0142	0.2451	0.0149	0.4034	0.0329
80	0.2225	0.0146	0.2453	0.0155	0.4042	0.0356
82	0.2215	0.0183	0.2455	0.0204	0.4022	0.0434
84	0.2362	0.1098	0.2566	0.0758	0.4128	0.0858
86	0.3039	0.1963	0.3451	0.2280	0.5354	0.3280
88	0.4972	0.3205	0.5978	0.3880	0.8964	0.5527
90	0.6503	0.3081	0.7785	0.3741	1.2132	0.4997
92	0.6882	0.2566	0.7883	0.3193	1.2174	0.4676
94	0.6866	0.2631	0.7831	0.2882	1.1846	0.4251
96	0.6806	0.2589	0.7903	0.3148	1.1817	0.4345
100	0.6841	0.2637	0.7795	0.3118	1.1774	0.4438
104	0.6549	0.2551	0.7351	0.2953	1.1427	0.4476
108	0.6501	0.2707	0.7303	0.2901	1.1068	0.4286
112	0.6206	0.2555	0.7134	0.2820	1.0498	0.4259
116	0.5756	0.2461	0.6829	0.2937	1.0066	0.3804
120	0.5376	0.2205	0.6463	0.2797	0.9703	0.4093
124	0.5115	0.2297	0.5890	0.2709	0.9249	0.3718
128	0.4642	0.2118	0.5133	0.2292	0.8114	0.3486
132	0.4247	0.1978	0.4667	0.2248	0.7388	0.3314
136	0.3465	0.1742	0.3902	0.2091	0.6430	0.3125
140	0.2819	0.1503	0.3309	0.1993	0.5014	0.2972
144	0.2137	0.1332	0.2285	0.1526	0.3691	0.2701
148	0.1589	0.1067	0.1803	0.1469	0.2508	0.2302
152	0.1289	0.0895	0.1346	0.1136	0.2005	0.2088

	r/R = 0.997		r/R = 0.995		r/R = 0.991	
θ deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
156	0.1143	0.0827	0.1098	0.0931	0.1533	0.1810
160	0.0905	0.0598	0.0868	0.0697	0.1076	0.1308
164	0.0694	0.0427	0.0608	0.0469	0.0692	0.0840
168	0.0498	0.0249	0.0437	0.0307	0.0356	0.0456
172	0.0370	0.0155	0.0269	0.0161	0.0158	0.0240
176	0.0310	0.0120	0.0208	0.0087	0.0069	0.0101
180	0.0318	0.0089	0.0228	0.0091	0.0081	0.0088

SPRE s/d = 30

	r/R = 0.986		r/R = 0.983		r/R = 0.977	
θ deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
4	0.0373	0.0301	0.0469	0.0385	0.0852	0.0658
8	0.0616	0.0285	0.0858	0.0445	0.1604	0.0744
12	0.1117	0.0296	0.1518	0.0422	0.2772	0.0730
16	0.1848	0.0358	0.2544	0.0547	0.4413	0.0861
20	0.2552	0.0367	0.3525	0.0488	0.5959	0.0703
24	0.3117	0.0384	0.4238	0.0568	0.6781	0.0526
28	0.3806	0.0451	0.5158	0.0579	0.7574	0.0548
32	0.4739	0.0501	0.6235	0.0478	0.8494	0.0565
36	0.5481	0.0517	0.6778	0.0353	0.9270	0.0579
40	0.5801	0.0446	0.7013	0.0404	0.9657	0.0584
44	0.6182	0.0400	0.7382	0.0449	1.0141	0.0606
48	0.6692	0.0361	0.7929	0.0429	1.0835	0.0608
52	0.6869	0.0370	0.8189	0.0439	1.1176	0.0627
56	0.6864	0.0366	0.8227	0.0441	1.1234	0.0589
60	0.7017	0.0381	0.8369	0.0437	1.1440	0.0624
64	0.7183	0.0386	0.8567	0.0463	1.1661	0.0618
68	0.7201	0.0376	0.8613	0.0473	1.1713	0.0612
72	0.6999	0.0363	0.8364	0.0445	1.1450	0.0599
76	0.6845	0.0339	0.8137	0.0416	1.1192	0.0573
78	0.6849	0.0334	0.8125	0.0416	1.1168	0.0556
80	0.6857	0.0354	0.8133	0.0445	1.1187	0.0598
82	0.6851	0.0468	0.8156	0.0597	1.1247	0.0934
84	0.7040	0.1422	0.8389	0.1726	1.1769	0.2684
86	0.8815	0.4617	1.0133	0.4917	1.4356	0.6394
88	1.2715	0.6818	1.4986	0.7660	1.9702	0.8691
90	1.6255	0.6443	1.8591	0.7161	2.3519	0.7936
92	1.6913	0.6071	1.9615	0.6624	2.4574	0.6771

θ	r/R = 0.986		r/R = 0.983		r/R = 0.977	
deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
94	1.6946	0.6074	1.9789	0.6213	2.4678	0.6807
96	1.6871	0.5502	1.9415	0.6171	2.4536	0.7117
100	1.6302	0.5521	1.9369	0.6550	2.3813	0.7080
104	1.6044	0.5642	1.8928	0.6416	2.3860	0.6878
108	1.5749	0.5591	1.8222	0.6244	2.2883	0.7423
112	1.5018	0.5265	1.7696	0.6222	2.2401	0.6850
116	1.4436	0.5522	1.6530	0.5737	2.1651	0.6862
120	1.3477	0.4865	1.6474	0.5719	2.0838	0.6271
124	1.3003	0.4609	1.5332	0.5467	1.9683	0.6284
128	1.2109	0.4660	1.3730	0.5039	1.8702	0.6015
132	1.0645	0.4153	1.3004	0.4861	1.6245	0.5544
136	0.9240	0.3925	1.1475	0.4545	1.4224	0.5479
140	0.7723	0.3924	0.9200	0.4202	1.2485	0.5131
144	0.5663	0.3610	0.6930	0.4222	0.9821	0.5237
148	0.4217	0.3586	0.5222	0.4133	0.7728	0.5108
152	0.3115	0.3009	0.4453	0.3798	0.5968	0.4878
156	0.2545	0.2666	0.3676	0.3346	0.5115	0.4205
160	0.1953	0.2150	0.2819	0.2781	0.4353	0.3711
164	0.1263	0.1614	0.1718	0.1952	0.2873	0.2819
168	0.0609	0.0990	0.0801	0.1219	0.1632	0.2026
172	0.0236	0.0460	0.0272	0.0558	0.0564	0.1073
176	0.0069	0.0191	0.0077	0.0212	0.0138	0.0321
180	0.0056	0.0112	0.0074	0.0140	0.0126	0.0214

SPRE s/d = 30

θ	r/R = 0.970		r/R = 0.950		r/R = 0.930	
deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
4	0.1244	0.0790	0.2039	0.1408	0.2477	0.1672
8	0.2371	0.1021	0.4289	0.1480	0.5320	0.1631
12	0.4010	0.0961	0.6672	0.1045	0.8054	0.1122
16	0.6112	0.0860	0.8972	0.0964	1.0759	0.1069
20	0.7477	0.0666	1.0986	0.0958	1.3290	0.1085
24	0.8447	0.0721	1.2469	0.0970	1.5345	0.1073
28	0.9495	0.0649	1.3963	0.0953	1.7012	0.1145
32	1.0587	0.0623	1.5541	0.1005	1.9112	0.1143
36	1.1475	0.0668	1.6977	0.1077	2.0948	0.1273
40	1.1992	0.0742	1.7769	0.1103	2.2052	0.1382
44	1.2577	0.0778	1.8667	0.1172	2.3149	0.1497

	r/R = 0.970		r/R = 0.950		r/R = 0.930	
θ deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
48	1.3438	0.0793	1.9718	0.1180	2.4561	0.1481
52	1.3922	0.0820	2.0501	0.1131	2.5581	0.1435
56	1.4033	0.0809	2.0791	0.1087	2.6032	0.1505
60	1.4237	0.0751	2.1183	0.1145	2.6492	0.1488
64	1.4489	0.0686	2.1580	0.1223	2.7047	0.1513
68	1.4570	0.0706	2.1672	0.1196	2.7382	0.1461
72	1.4283	0.0701	2.1334	0.1131	2.7073	0.1473
76	1.3961	0.0684	2.0979	0.1053	2.6776	0.1480
78	1.3915	0.0701	2.0936	0.1053	2.6813	0.1536
80	1.3911	0.0754	2.0981	0.1211	2.6920	0.1806
82	1.3957	0.1186	2.1212	0.2103	2.7545	0.3238
84	1.4481	0.2883	2.2022	0.4088	2.8287	0.4909
86	1.7108	0.6697	2.4713	0.7212	3.0642	0.7348
88	2.2563	0.8921	2.9321	0.8852	3.2940	0.8319
90	2.6635	0.7983	3.2867	0.7776	3.5876	0.7432
92	2.7510	0.6934	3.4568	0.7104	3.7206	0.6784
94	2.8193	0.7008	3.4468	0.6715	3.8405	0.6174
96	2.8312	0.7270	3.4126	0.7078	3.8389	0.6551
100	2.8039	0.7302	3.4824	0.6817	3.8441	0.6419
104	2.7562	0.7352	3.4325	0.7138	3.7617	0.6016
108	2.6639	0.7039	3.3806	0.7209	3.6617	0.6342
112	2.5983	0.6966	3.2049	0.6778	3.5856	0.6383
116	2.5513	0.6828	3.1253	0.6559	3.4889	0.6414
120	2.3965	0.6836	3.0126	0.6672	3.3386	0.6138
124	2.2665	0.6244	2.9013	0.6735	3.1638	0.5978
128	2.1682	0.6152	2.7201	0.6187	3.0381	0.5997
132	1.9850	0.5949	2.5835	0.6179	2.8437	0.5838
136	1.7621	0.5524	2.3565	0.6043	2.6191	0.5579
140	1.5342	0.5730	2.0324	0.5706	2.3508	0.5599
144	1.2015	0.5604	1.6923	0.5912	1.9875	0.5721
148	1.0081	0.5689	1.4102	0.5976	1.6988	0.5895
152	0.7894	0.5253	1.1703	0.6278	1.4102	0.5920
156	0.7077	0.4783	1.0212	0.5946	1.2655	0.5765
160	0.6147	0.4242	0.9088	0.5068	1.0895	0.5451
164	0.4377	0.3447	0.7367	0.4132	0.8717	0.4540
168	0.2616	0.2701	0.4860	0.3343	0.6221	0.3879
172	0.1106	0.1727	0.2615	0.2753	0.3868	0.3152
176	0.0268	0.0741	0.0800	0.1375	0.1451	0.1868
180	0.0187	0.0296	0.0358	0.0573	0.0407	0.0597

SPRE s/d = 30

θ	r/R = 0.863		r/R = 0.730		r/R = 0.597	
deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
4	0.1897	0.1758	0.1271	0.1477	0.1026	0.1369
8	0.5612	0.1980	0.4766	0.2026	0.4384	0.2035
12	0.9225	0.1467	0.8797	0.1478	0.8386	0.1415
16	1.2787	0.1312	1.2826	0.1306	1.2404	0.1306
20	1.6237	0.1134	1.6631	0.1207	1.6287	0.1225
24	1.8994	0.1110	1.9963	0.1017	1.9563	0.1025
28	2.1925	0.0992	2.3145	0.0940	2.2943	0.0945
32	2.4811	0.1054	2.6627	0.0801	2.6494	0.0830
36	2.7599	0.1138	2.9949	0.0639	2.9966	0.0743
40	2.9580	0.1309	3.2665	0.0789	3.2854	0.0763
44	3.1574	0.1445	3.5312	0.0661	3.5637	0.0618
48	3.3837	0.1595	3.8185	0.0877	3.8749	0.0744
52	3.5440	0.1626	4.0665	0.0845	4.1332	0.0648
56	3.6648	0.1599	4.2638	0.0978	4.3637	0.0589
60	3.7640	0.1852	4.4548	0.1104	4.5771	0.0519
64	3.8815	0.1883	4.6565	0.0960	4.7909	0.0634
68	3.9599	0.1934	4.8231	0.1309	5.0007	0.0537
72	3.9941	0.2028	4.9495	0.1237	5.1572	0.0689
76	4.0167	0.2120	5.0540	0.1503	5.3080	0.0796
78	4.0474	0.2348	5.1188	0.1961	5.3410	0.1123
80	4.1055	0.3165	5.1509	0.2516	5.3593	0.1681
82	4.1659	0.4514	5.1571	0.3271	5.3713	0.2596
84	4.2555	0.5667	5.0901	0.4380	5.3757	0.3253
86	4.2060	0.6791	4.9869	0.5268	5.2971	0.4407
88	4.1898	0.6417	4.7795	0.5685	5.1334	0.4996
90	4.1323	0.6144	4.6516	0.5322	5.0623	0.5127
92	4.2353	0.5335	4.6917	0.4916	4.9664	0.4429
94	4.2621	0.5185	4.7140	0.4528	5.0026	0.4524
96	4.2915	0.5127	4.7246	0.4272	5.0379	0.4072
100	4.2946	0.5104	4.7360	0.4256	5.0424	0.4070
104	4.2471	0.5052	4.6694	0.4629	4.9469	0.3999
108	4.1281	0.5151	4.5519	0.4556	4.8433	0.4081
112	4.0263	0.5221	4.4635	0.4406	4.7614	0.4292
116	3.9331	0.5159	4.3388	0.4337	4.6459	0.4107
120	3.8360	0.5048	4.2186	0.4224	4.5371	0.3885
124	3.6852	0.4614	4.0709	0.4149	4.3846	0.3707
128	3.4594	0.4555	3.8938	0.4102	4.2170	0.3413
132	3.3623	0.4358	3.7554	0.3608	4.0335	0.3542
136	3.0952	0.4600	3.5720	0.3755	3.8815	0.3448
140	2.8697	0.4463	3.3432	0.3992	3.7086	0.3261
144	2.5185	0.4571	3.0977	0.4084	3.4918	0.3349
148	2.2551	0.4992	2.8749	0.3936	3.2381	0.3486
152	1.9521	0.4861	2.5834	0.3949	2.8854	0.3934

	r/R = 0.863		r/R = 0.730		r/R = 0.597	
θ deg.	ū m/sec	u' m/sec	ū m/sec	u' m/sec	ū m/sec	u' m/sec
156	1.7407	0.4982	2.2772	0.4245	2.5686	0.4031
160	1.5122	0.4608	1.9568	0.4098	2.2493	0.3629
164	1.2595	0.4256	1.5218	0.3567	1.8647	0.3313
168	0.9339	0.3594	1.1947	0.3620	1.4303	0.3310
172	0.6254	0.3205	0.8478	0.3237	1.0158	0.3207
176	0.3014	0.2606	0.4738	0.2886	0.6024	0.2804
180	0.0743	0.1096	0.1391	0.1576	0.1890	0.1798

SPRE s/d = 30

	r/R = 0.463		r/R = 0.330		r/R = 0.197	
θ deg.	ū m/sec	u' m/sec	ū m/sec	u' m/sec	ū m/sec	u' m/sec
4	0.0971	0.1316	0.0732	0.1066	0.0687	0.1002
8	0.4091	0.2016	0.3531	0.1962	0.3376	0.1789
12	0.8217	0.1481	0.7710	0.1438	0.7600	0.1406
16	1.2122	0.1318	1.1743	0.1329	1.1562	0.1275
20	1.5890	0.1211	1.5518	0.1171	1.5440	0.1145
24	1.9259	0.1094	1.8953	0.1008	1.8825	0.0975
28	2.2617	0.1001	2.2262	0.0946	2.2219	0.0894
32	2.6275	0.0880	2.5827	0.0785	2.5886	0.0781
36	2.9802	0.0821	2.9439	0.0832	2.9378	0.0714
40	3.2799	0.0839	3.2453	0.0695	3.2476	0.0713
44	3.5592	0.0671	3.5282	0.0619	3.5302	0.0564
48	3.8774	0.0735	3.8452	0.0715	3.8505	0.0680
52	4.1476	0.0676	4.1218	0.0586	4.1266	0.0557
56	4.3834	0.0490	4.3662	0.0508	4.3699	0.0430
60	4.6030	0.0543	4.5780	0.0519	4.5938	0.0464
64	4.8273	0.0519	4.8073	0.0519	4.8239	0.0432
68	5.0343	0.0471	5.0181	0.0367	5.0317	0.0384
72	5.2086	0.0412	5.1995	0.0393	5.2076	0.0271
76	5.3488	0.0594	5.3285	0.0635	5.3346	0.0602
78	5.3835	0.0836	5.3719	0.0718	5.3848	0.0701
80	5.4358	0.1228	5.4203	0.1020	5.4381	0.0933
82	5.4636	0.1797	5.4855	0.1430	5.5006	0.1362
84	5.5027	0.2541	5.5259	0.2004	5.5880	0.1829
86	5.5492	0.3527	5.6553	0.2932	5.7475	0.2735
88	5.4658	0.4258	5.7252	0.3503	5.8348	0.3277
90	5.3750	0.4176	5.5189	0.3882	5.7121	0.3678
92	5.2904	0.3866	5.4458	0.3606	5.5736	0.3235

	r/R = 0.463		r/R = 0.330		r/R = 0.197	
θ deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
94	5.2671	0.4079	5.4283	0.3132	5.5174	0.2973
96	5.3012	0.3708	5.4024	0.3191	5.4866	0.2845
100	5.2760	0.3768	5.3812	0.3190	5.4673	0.2997
104	5.1828	0.3757	5.3070	0.3541	5.4086	0.2862
108	5.0290	0.3864	5.1944	0.3462	5.2862	0.3070
112	4.9625	0.3895	5.1068	0.3668	5.1984	0.3145
116	4.8367	0.3923	5.0279	0.3377	5.1315	0.3147
120	4.7924	0.3407	4.9284	0.3276	5.0588	0.2925
124	4.6349	0.3370	4.7958	0.2837	4.8917	0.2485
128	4.4514	0.3171	4.6152	0.2476	4.6810	0.2168
132	4.3041	0.2808	4.3995	0.2225	4.4778	0.1800
136	4.0874	0.2718	4.1884	0.2123	4.2553	0.1516
140	3.9011	0.2420	3.9631	0.1577	3.9988	0.1468
144	3.6859	0.2470	3.7222	0.2011	3.7647	0.1594
148	3.3989	0.2934	3.4945	0.1822	3.5291	0.1844
152	3.1349	0.2861	3.2133	0.2306	3.2556	0.1904
156	2.8028	0.3192	2.8927	0.2416	2.9355	0.2186
160	2.4692	0.3168	2.5394	0.2526	2.5917	0.2230
164	2.0728	0.2965	2.1477	0.2668	2.1986	0.2524
168	1.6280	0.2919	1.7119	0.2847	1.7543	0.2438
172	1.1711	0.3228	1.2445	0.2866	1.2853	0.2922
176	0.7309	0.3033	0.7913	0.2683	0.8241	0.2693
180	0.2771	0.2263	0.3059	0.2227	0.3267	0.2293

SPRE s/d = 30

	r/R = 0.004	
θ deg.	ũ m/sec	u' m/sec
4	0.0483	0.0823
8	0.2880	0.1821
12	0.7132	0.1397
16	1.1214	0.1212
20	1.5222	0.1050
24	1.8684	0.0920
28	2.2172	0.0854
32	2.5743	0.0757
36	2.9413	0.0733
40	3.2357	0.0684
44	3.5223	0.0591

r/R = 0.004

θ deg.	ū m/sec	u' m/sec
48	3.8386	0.0697
52	4.1161	0.0579
56	4.3661	0.0433
60	4.5912	0.0489
64	4.8221	0.0421
68	5.0350	0.0436
72	5.2086	0.0380
76	5.3317	0.0587
78	5.3824	0.0699
80	5.4434	0.0999
82	5.5189	0.1243
84	5.6274	0.1967
86	5.8429	0.2721
88	5.9452	0.2875
90	5.7876	0.3223
92	5.6441	0.2749
94	5.5613	0.2563
96	5.5436	0.2658
100	5.4992	0.2680
104	5.4438	0.2642
108	5.3695	0.2581
112	5.2783	0.2748
116	5.2111	0.2605
120	5.0778	0.2645
124	4.9092	0.2337
128	4.6908	0.2052
132	4.4353	0.1898
136	4.1803	0.2067
140	3.9119	0.2114
144	3.6736	0.2293
148	3.3997	0.2214
152	3.0768	0.2488
156	2.7157	0.2850
160	2.4331	0.2750
164	2.0180	0.2819
168	1.5719	0.2865
172	1.1803	0.2927
176	0.7590	0.2601
180	0.2964	0.2264

SPRE s/d = 30

θ deg.	u_m m/sec
2	0.0650
4	0.1289
6	0.2573
8	0.4300
10	0.6140
12	0.7884
14	0.9614
16	1.1425
18	1.3179
20	1.4791
22	1.6229
24	1.7654
26	1.9026
28	2.0498
30	2.2011
32	2.3539
34	2.5058
36	2.6461
38	2.7681
40	2.8776
42	2.9878
44	3.1017
46	3.2278
48	3.3548
50	3.4646
52	3.5617
54	3.6501
56	3.7302
58	3.8031
60	3.8860
62	3.9661
64	4.0506
66	4.1306
68	4.1919
70	4.2457
72	4.2915
74	4.3252
76	4.3718
78	4.4064
80	4.4411
82	4.4743
84	4.5037
86	4.5366
88	4.5397
90	4.5178
92	4.5232
94	4.5396
96	4.5506
98	4.5396
100	4.5480
102	4.5195
104	4.4788
106	4.4218
108	4.3683
110	4.3209
112	4.2819
114	4.2177
116	4.1812
118	4.1383
120	4.0805
122	4.0074
124	3.9339
126	3.8431
128	3.7566
130	3.6666
132	3.5998
134	3.4890
136	3.3942
138	3.2847
140	3.1760
142	3.0565
144	2.9107
146	2.7914
148	2.6639
150	2.5169
152	2.3798
154	2.2533
156	2.1192
158	1.9760
160	1.8491
162	1.6835
164	1.5165
166	1.3386
168	1.1669
170	0.9951
172	0.8192
174	0.6515
176	0.4711
178	0.3045
180	0.1588

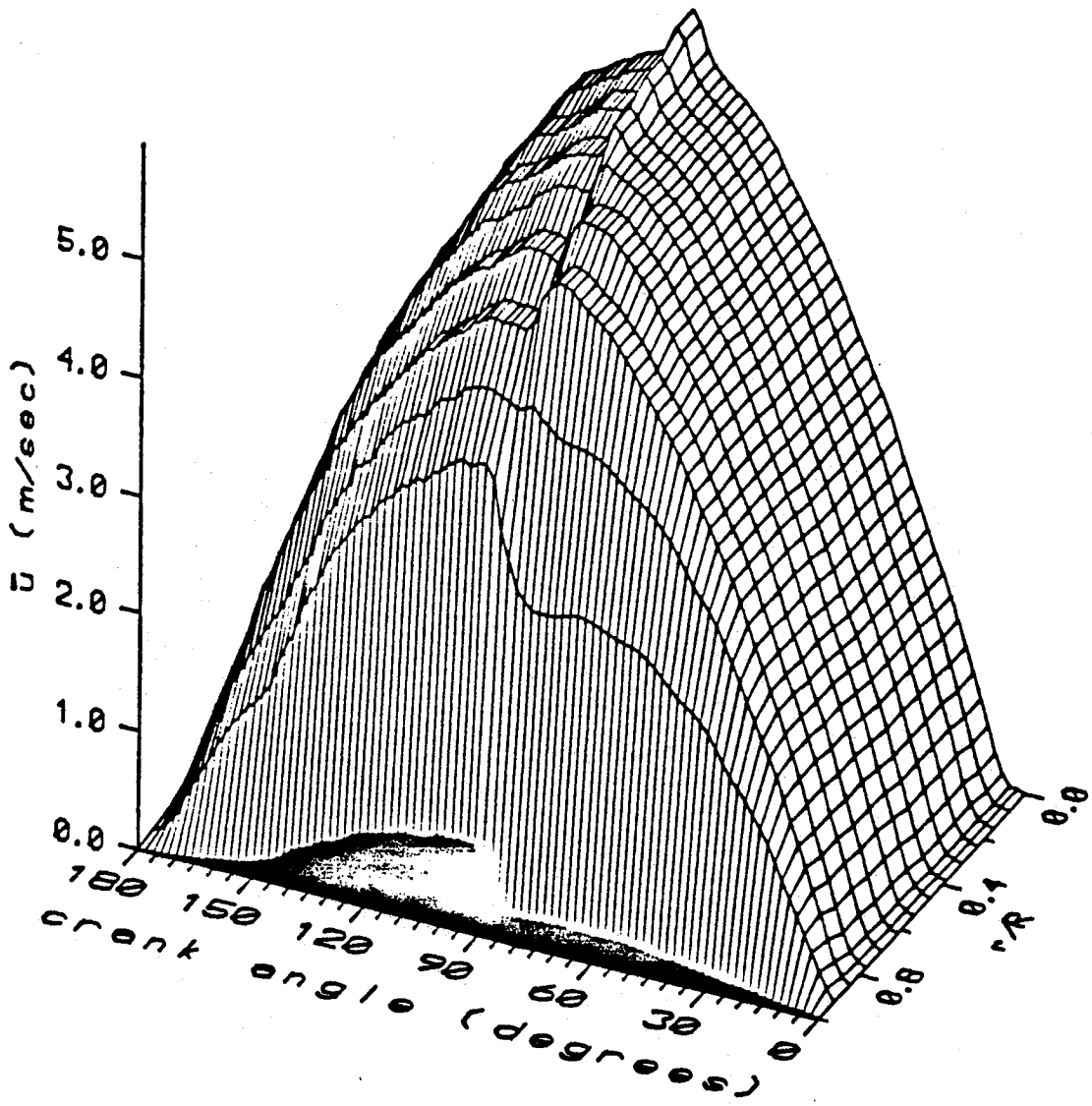


Figure 7: Ensemble-averaged velocity at s/d = 30

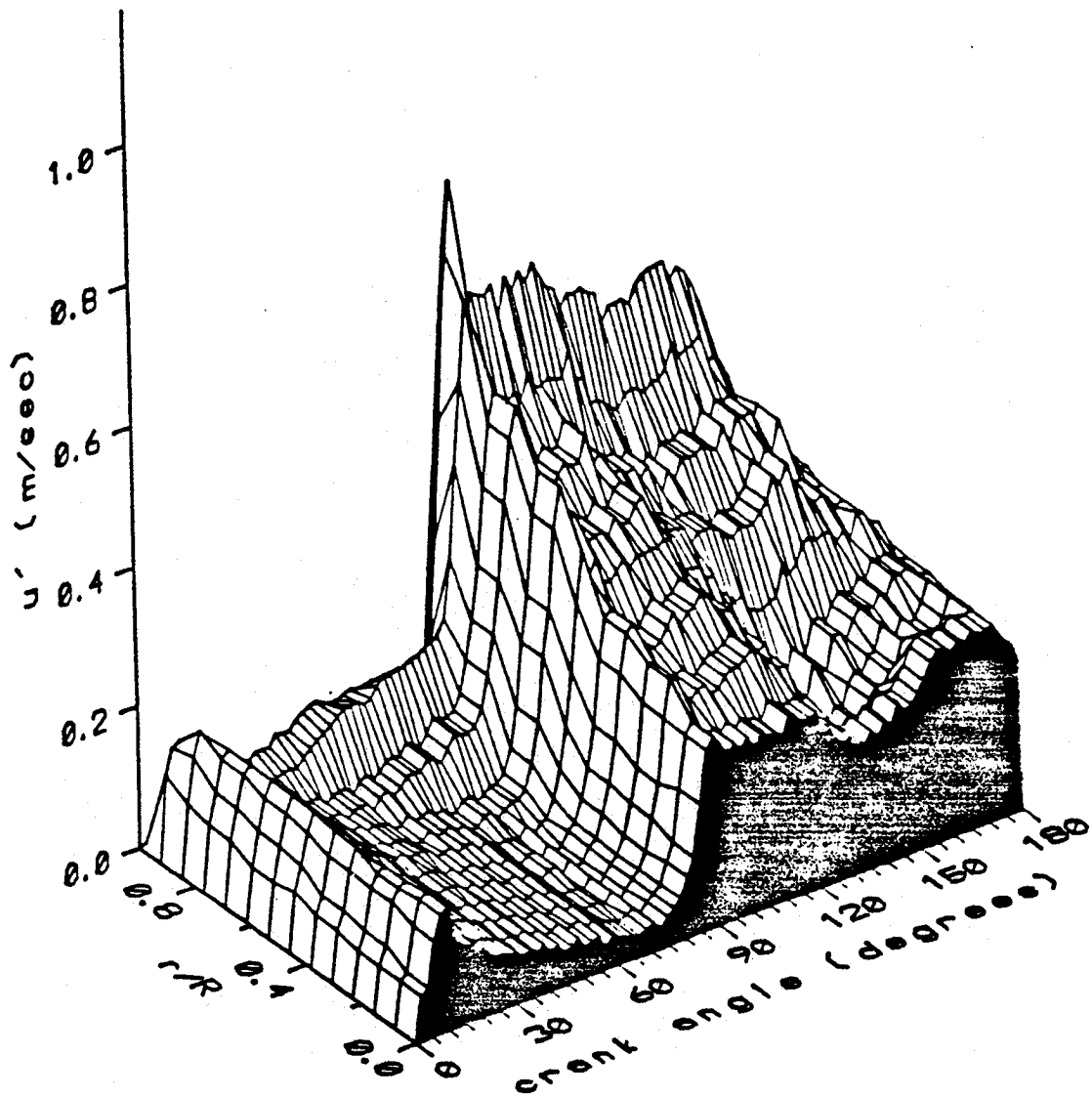


Figure 8: Streamwise velocity fluctuation at s/d = 30
Centerline view

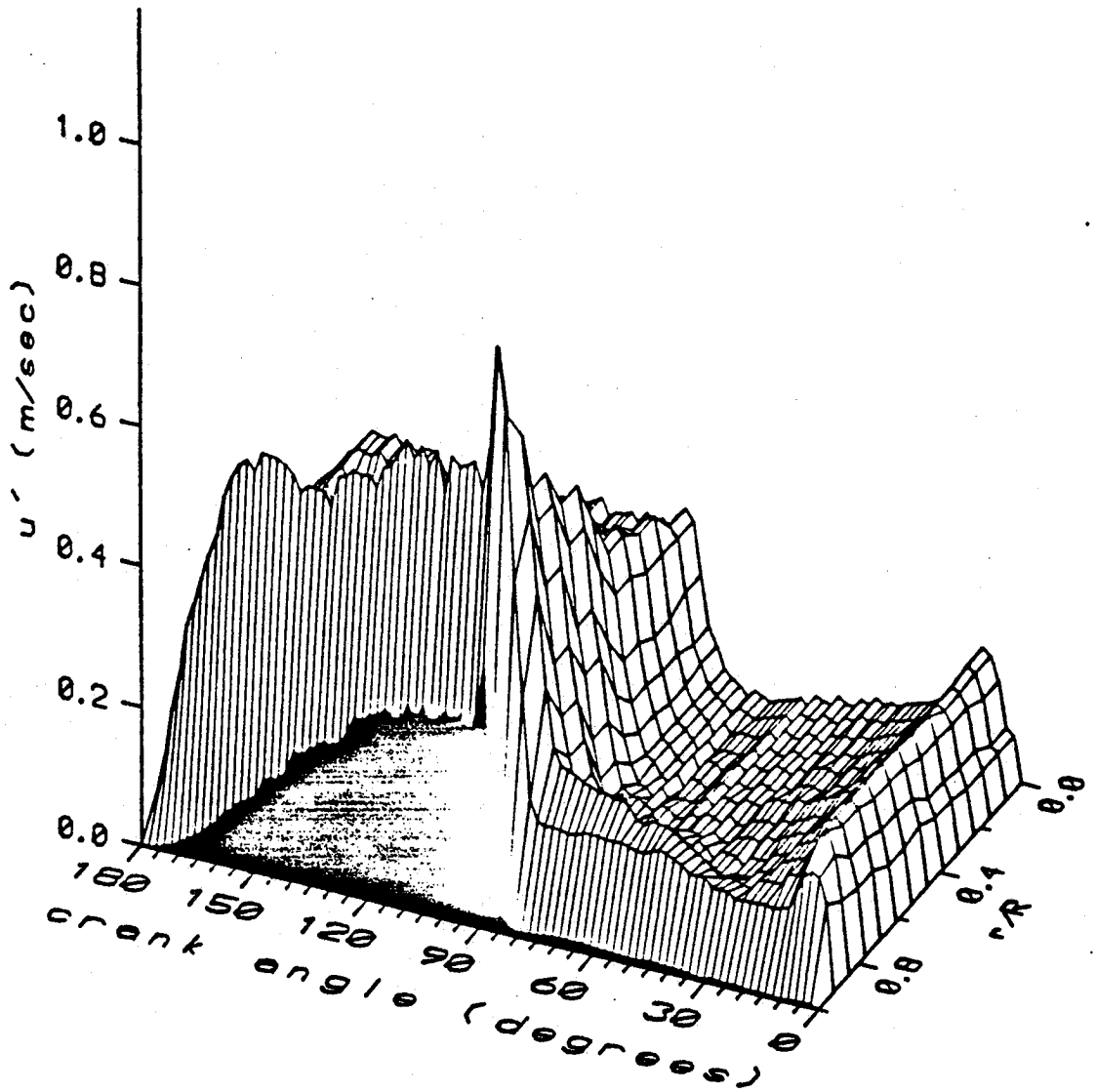


Figure 9: Streamwise velocity fluctuation at s/d = 30
Near-wall view

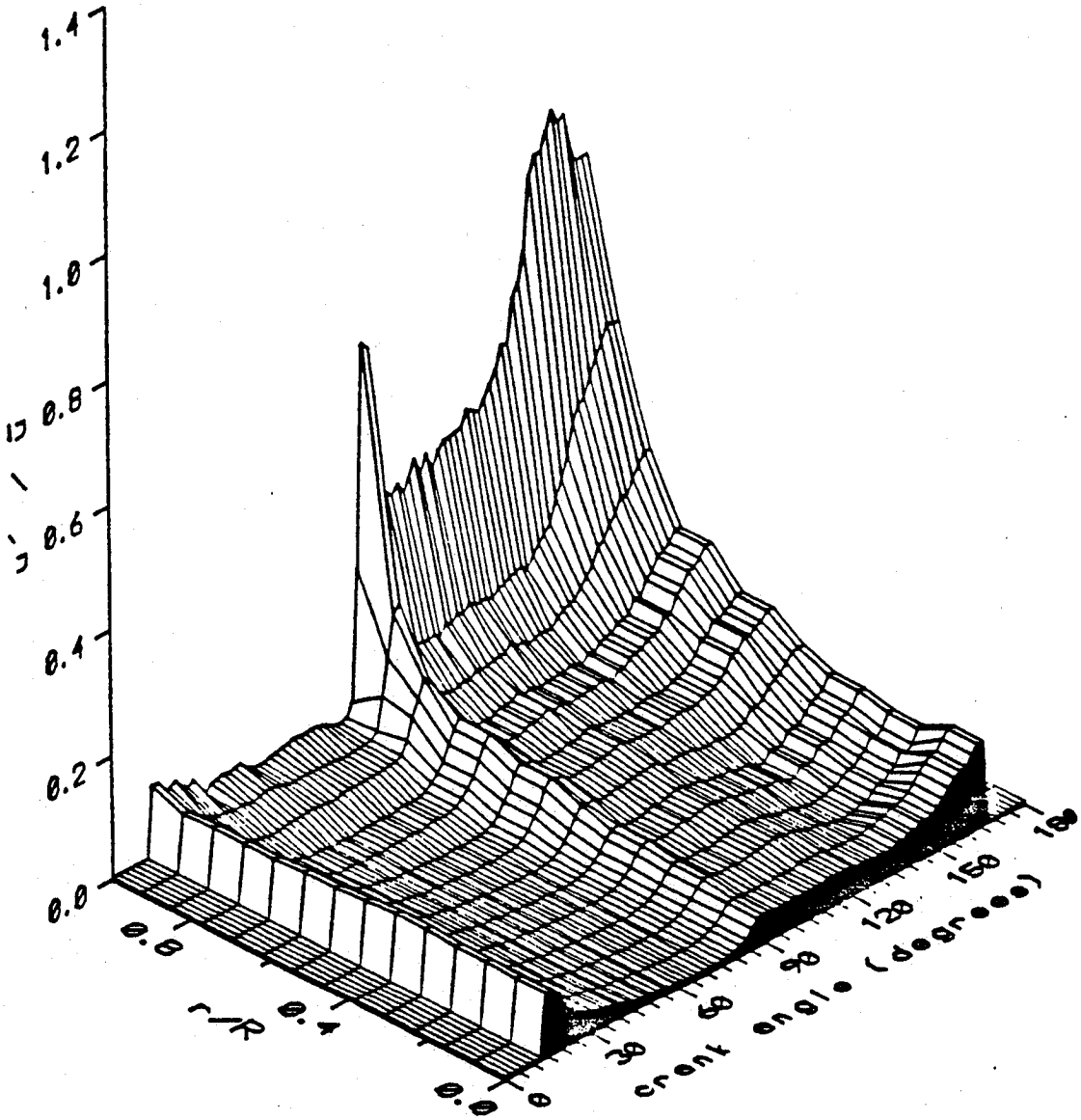


Figure 10: Streamwise turbulence intensity, u' / ũ , at s/d = 30

Note: The peaks appear lower than the actual data due to smoothing by the plotting package. The peak turbulence intensity is 1.25, at 164° for r/R = 0.986.

SPRE s/d = 44

θ	r/R = 0.999		r/R = 0.998		r/R = 0.995	
deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
4	0.0679	0.0169	0.0394	0.0138	0.0181	0.0168
8	0.0885	0.0151	0.0598	0.0170	0.0410	0.0201
12	0.1145	0.0199	0.0935	0.0230	0.0848	0.0377
16	0.1376	0.0148	0.1130	0.0189	0.1150	0.0305
20	0.1429	0.0135	0.1223	0.0151	0.1232	0.0228
24	0.1522	0.0122	0.1308	0.0146	0.1389	0.0224
28	0.1633	0.0091	0.1450	0.0148	0.1604	0.0194
32	0.1749	0.0109	0.1647	0.0118	0.1901	0.0219
36	0.1875	0.0106	0.1750	0.0132	0.2098	0.0217
40	0.1899	0.0102	0.1797	0.0131	0.2172	0.0213
44	0.1968	0.0099	0.1898	0.0134	0.2318	0.0195
48	0.2150	0.0122	0.2106	0.0153	0.2637	0.0235
52	0.2199	0.0114	0.2178	0.0137	0.2747	0.0221
56	0.2194	0.0108	0.2156	0.0140	0.2697	0.0227
60	0.2229	0.0097	0.2211	0.0123	0.2809	0.0218
64	0.2305	0.0067	0.2301	0.0098	0.2999	0.0199
68	0.2315	0.0064	0.2312	0.0091	0.3011	0.0190
72	0.2255	0.0081	0.2235	0.0100	0.2840	0.0196
76	0.2247	0.0087	0.2214	0.0107	0.2783	0.0194
80	0.2252	0.0080	0.2243	0.0098	0.2847	0.0174
84	0.2215	0.0091	0.2164	0.0116	0.2694	0.0188
88	0.2060	0.0105	0.1994	0.0106	0.2449	0.0149
92	0.1986	0.0078	0.1910	0.0084	0.2327	0.0124
96	0.1953	0.0070	0.1869	0.0096	0.2268	0.0132
98	0.1924	0.0062	0.1825	0.0103	0.2211	0.0149
100	0.1887	0.0085	0.1766	0.0111	0.2126	0.0186
102	0.1913	0.0491	0.1826	0.0564	0.2112	0.0509
104	0.2424	0.1590	0.2549	0.1910	0.3006	0.2537
106	0.3962	0.2483	0.5145	0.3567	0.6439	0.4667
108	0.5324	0.2237	0.6467	0.2803	0.8915	0.4358
110	0.5551	0.1949	0.6454	0.2625	0.8907	0.3570
112	0.5107	0.1917	0.5990	0.2353	0.8210	0.3077
114	0.5041	0.1830	0.5345	0.2224	0.7854	0.3078
116	0.4791	0.1796	0.5499	0.2339	0.7430	0.2945
120	0.4520	0.1747	0.5205	0.2097	0.7248	0.3055
124	0.4275	0.1641	0.4888	0.2135	0.6820	0.2944
128	0.3805	0.1546	0.4320	0.1871	0.6063	0.2814
132	0.3639	0.1451	0.3944	0.1796	0.5307	0.2398
136	0.3162	0.1311	0.3397	0.1647	0.4722	0.2415
140	0.2832	0.1135	0.2922	0.1453	0.4088	0.2155
144	0.2388	0.0828	0.2479	0.1210	0.3345	0.1945
148	0.2010	0.0707	0.2067	0.1023	0.2408	0.1476
152	0.1675	0.0573	0.1611	0.0751	0.1847	0.1256

θ	r/R = 0.999		r/R = 0.998		r/R = 0.995	
deg.	ū m/sec	u' m/sec	ū m/sec	u' m/sec	ū m/sec	u' m/sec
156	0.1508	0.0517	0.1234	0.0626	0.1241	0.0885
160	0.1187	0.0351	0.0920	0.0403	0.0826	0.0630
164	0.0926	0.0241	0.0641	0.0279	0.0459	0.0353
168	0.0754	0.0183	0.0458	0.0166	0.0257	0.0205
172	0.0639	0.0105	0.0361	0.0078	0.0141	0.0101
176	0.0613	0.0070	0.0341	0.0050	0.0112	0.0052
180	0.0669	0.0080	0.0393	0.0064	0.0161	0.0061

SPRE s/d = 44

θ	r/R = 0.993		r/R = 0.990		r/R = 0.983	
deg.	ū m/sec	u' m/sec	ū m/sec	u' m/sec	ū m/sec	u' m/sec
4	0.0120	0.0170	0.0132	0.0252	0.0170	0.0425
8	0.0372	0.0268	0.0401	0.0291	0.0902	0.0840
12	0.0976	0.0508	0.1057	0.0490	0.2657	0.1313
16	0.1435	0.0434	0.1635	0.0569	0.3847	0.1155
20	0.1611	0.0354	0.1774	0.0447	0.4358	0.1021
24	0.1797	0.0353	0.1999	0.0394	0.5069	0.0943
28	0.2097	0.0352	0.2362	0.0381	0.5867	0.0807
32	0.2575	0.0341	0.3016	0.0404	0.6783	0.0610
36	0.2907	0.0362	0.3309	0.0382	0.7215	0.0578
40	0.3022	0.0335	0.3466	0.0372	0.7398	0.0523
44	0.3258	0.0331	0.3750	0.0375	0.7732	0.0500
48	0.3697	0.0318	0.4249	0.0382	0.8384	0.0490
52	0.3857	0.0317	0.4451	0.0396	0.8674	0.0491
56	0.3817	0.0323	0.4405	0.0349	0.8576	0.0479
60	0.3962	0.0323	0.4587	0.0373	0.9767	0.0468
64	0.4215	0.0312	0.4950	0.0415	0.9100	0.0475
68	0.4265	0.0308	0.4945	0.0401	0.9155	0.0456
72	0.4012	0.0325	0.4687	0.0369	0.8858	0.0435
76	0.3890	0.0303	0.4599	0.0333	0.8731	0.0404
80	0.3976	0.0287	0.4722	0.0344	0.8859	0.0392
84	0.3784	0.0252	0.4448	0.0304	0.8613	0.0383
88	0.3487	0.0213	0.4092	0.0260	0.8163	0.0358
92	0.3275	0.0221	0.3843	0.0291	0.7852	0.0350
96	0.3155	0.0217	0.3690	0.0297	0.7663	0.0361
98	0.3052	0.0218	0.3550	0.0287	0.7513	0.0379
100	0.2944	0.0263	0.3416	0.0316	0.7327	0.0419
102	0.2979	0.0887	0.3458	0.1358	0.7365	0.1261
104	0.4547	0.4180	0.4528	0.3597	0.9042	0.4893

	r/R = 0.993		r/R = 0.990		r/R = 0.983	
θ deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
106	0.8857	0.6026	0.9176	0.6649	1.5414	0.8430
108	1.1421	0.5167	1.2536	0.5847	2.0079	0.7134
110	1.1625	0.4379	1.2336	0.4619	1.9946	0.6403
112	1.0661	0.3901	1.1672	0.4382	1.8954	0.6097
114	0.9992	0.3930	1.1289	0.4413	1.7707	0.5622
116	0.9804	0.3759	1.0503	0.4384	1.7904	0.6326
120	0.9366	0.3824	1.0411	0.4056	1.7114	0.5993
124	0.8601	0.3752	0.9290	0.3785	1.6133	0.5513
128	0.7832	0.3364	0.8652	0.3637	1.5037	0.5277
132	0.7115	0.3430	0.7741	0.3312	1.3089	0.4739
136	0.6264	0.2907	0.6980	0.3144	1.2445	0.4469
140	0.5450	0.2882	0.5878	0.3066	1.1051	0.4276
144	0.4543	0.2542	0.4946	0.2581	0.9531	0.3917
148	0.3452	0.2209	0.3784	0.2296	0.7641	0.3678
152	0.2398	0.1676	0.2821	0.1882	0.5678	0.3223
156	0.1605	0.1305	0.2048	0.1691	0.4294	0.2941
160	0.1022	0.0922	0.1165	0.1087	0.2684	0.2334
164	0.0464	0.0523	0.0558	0.0742	0.1431	0.1627
168	0.0199	0.0239	0.0184	0.0253	0.0494	0.0798
172	0.0073	0.0106	0.0058	0.0104	0.0131	0.0324
176	0.0046	0.0043	0.0035	0.0059	0.0027	0.0071
180	0.0091	0.0061	0.0079	0.0063	0.0080	0.0096

SPRE s/d = 44

	r/R = 0.977		r/R = 0.937		r/R = 0.870	
θ deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
4	0.0324	0.0662	0.1517	0.1868	0.1760	0.2203
8	0.1675	0.1230	0.5666	0.2027	0.6532	0.2772
12	0.4471	0.1809	0.9525	0.2115	1.1394	0.2322
16	0.6054	0.1335	1.1330	0.1637	1.3798	0.1818
20	0.6874	0.0947	1.3518	0.1562	1.7091	0.1466
24	0.7557	0.0932	1.5421	0.1628	2.0057	0.1321
28	0.8294	0.0965	1.6942	0.1627	2.2698	0.1378
32	0.9292	0.0867	1.8833	0.1605	2.5607	0.1370
36	0.9904	0.0775	2.0363	0.1575	2.8223	0.1507
40	1.0278	0.0728	2.1139	0.1576	3.0111	0.1564
44	1.0667	0.0710	2.2146	0.1585	3.2037	0.1615
48	1.1458	0.0675	2.3611	0.1539	3.4128	0.1782
52	1.1826	0.0675	2.4481	0.1378	3.5804	0.1869

	r/R = 0.977		r/R = 0.937		r/R = 0.870	
θ deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
56	1.1826	0.0665	2.4660	0.1360	3.6999	0.1829
60	1.2025	0.0645	2.4977	0.1303	3.8072	0.1882
64	1.2392	0.0626	2.5737	0.1307	3.9256	0.1926
68	1.2522	0.0629	2.6085	0.1262	4.0121	0.1994
72	1.2175	0.0558	2.5874	0.1224	4.0438	0.1922
76	1.2023	0.0528	2.5762	0.1192	4.0637	0.1876
80	1.2133	0.0529	2.5989	0.1130	4.1087	0.1756
84	1.1858	0.0519	2.5755	0.1106	4.1202	0.1748
88	1.1302	0.0523	2.4952	0.1095	4.0717	0.1825
92	1.0886	0.0493	2.4268	0.1141	4.0150	0.1885
96	1.0643	0.0452	2.3809	0.1171	3.9688	0.2075
98	1.0461	0.0459	2.3584	0.1284	3.9527	0.2676
100	1.0257	0.0564	2.3458	0.1863	3.9960	0.3967
102	1.0181	0.1077	2.4015	0.4307	4.0803	0.5719
104	1.2060	0.5633	2.5811	0.6970	4.0503	0.7338
106	1.8435	0.9388	3.0566	0.8900	4.0327	0.7855
108	2.4041	0.8467	3.4520	0.7341	4.1308	0.6442
110	2.5028	0.6670	3.6132	0.6264	4.1439	0.5615
112	2.4170	0.6826	3.5184	0.6255	4.1458	0.4927
114	2.3022	0.6674	3.5371	0.6340	4.0591	0.5401
116	2.2432	0.6729	3.4228	0.6203	4.0482	0.5029
120	2.2070	0.6548	3.2797	0.6412	3.8741	0.5011
124	2.0683	0.6151	3.1391	0.6368	3.7211	0.4816
128	1.9162	0.5940	3.0258	0.5923	3.5924	0.4979
132	1.7515	0.5639	2.8029	0.5749	3.3636	0.4703
136	1.6410	0.5497	2.6442	0.5516	3.2182	0.4544
140	1.4515	0.4973	2.4764	0.5440	2.9711	0.4452
144	1.2762	0.4863	2.2044	0.5283	2.7366	0.4145
148	1.1000	0.4286	1.9766	0.4827	2.4576	0.4157
152	0.8741	0.4065	1.7075	0.4639	2.1639	0.4141
156	0.6715	0.3726	1.4717	0.4135	1.9133	0.3663
160	0.4760	0.3132	1.1796	0.3852	1.5824	0.3563
164	0.2585	0.2523	0.8549	0.3689	1.2611	0.3266
168	0.1268	0.1635	0.5624	0.3275	0.8801	0.3096
172	0.0297	0.0592	0.2632	0.2537	0.5481	0.2928
176	0.0062	0.0135	0.0595	0.0968	0.2069	0.2039
180	0.0172	0.0197	0.0401	0.0524	0.0316	0.0524

SPRE s/d = 44

θ	r/R = 0.737		r/R = 0.603		r/R = 0.470	
deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
4	0.0885	0.0315	0.0070	0.0105	0.0050	0.0078
8	0.0534	0.0850	0.0131	0.0276	0.0290	0.0590
12	1.0468	0.2725	0.6281	0.3728	0.2761	0.2824
16	1.3727	0.1959	1.2855	0.2300	1.1794	0.2269
20	1.7895	0.1390	1.7556	0.1403	1.7037	0.1484
24	2.0970	0.1156	2.0925	0.1091	2.0709	0.1112
28	2.4110	0.1002	2.4119	0.0979	2.3924	0.1014
32	2.7576	0.0968	2.7561	0.0926	2.7581	0.0911
36	3.0909	0.0828	3.0961	0.0800	3.0888	0.0770
40	3.3628	0.0776	3.3824	0.0695	3.3816	0.0597
44	3.6138	0.0788	3.6493	0.0617	3.6497	0.0590
48	3.9276	0.0748	3.9768	0.0571	3.9765	0.0562
52	4.2056	0.0862	4.2753	0.0570	4.2845	0.0580
56	4.3954	0.0994	4.4960	0.0584	4.5070	0.0464
60	4.5703	0.1078	4.6900	0.0584	4.7205	0.0564
64	4.7825	0.1162	4.9231	0.0571	4.9566	0.0617
68	4.9590	0.1221	5.1353	0.0660	5.1808	0.0518
72	5.0682	0.1279	5.2695	0.0574	5.2871	0.0486
76	5.1747	0.1307	5.3754	0.0599	5.4235	0.0586
80	5.2762	0.1309	5.5219	0.0762	5.5754	0.0346
84	5.3544	0.1390	5.6261	0.0587	5.6841	0.0574
88	5.3583	0.1445	5.6593	0.0688	5.7430	0.0447
92	5.3583	0.1598	5.7072	0.0823	5.7922	0.0484
96	5.4148	0.2008	5.7574	0.1389	5.8401	0.0994
98	5.4177	0.2614	5.7388	0.1927	5.8261	0.1308
100	5.4058	0.3432	5.6761	0.2335	5.7786	0.1877
102	5.3163	0.4190	5.5796	0.3126	5.7355	0.2545
104	5.0815	0.6030	5.4226	0.4548	5.6615	0.3645
106	4.7672	0.6295	5.2393	0.5362	5.6126	0.4604
108	4.6493	0.5425	5.0045	0.5013	5.3347	0.4807
110	4.6082	0.4752	4.8748	0.4402	5.1652	0.3988
112	4.5306	0.4079	4.8374	0.4214	5.0798	0.3617
114	4.4953	0.4181	4.7938	0.3767	5.0222	0.3457
116	4.4253	0.3997	4.7369	0.3415	4.9318	0.3364
120	4.3010	0.4249	4.5980	0.3610	4.8179	0.3370
124	4.1668	0.4124	4.4165	0.3873	4.6565	0.3266
128	3.9532	0.3680	4.2294	0.3672	4.4694	0.3168
132	3.7664	0.3873	4.0485	0.3623	4.2283	0.3114
136	3.5444	0.3910	3.8037	0.3206	4.0040	0.3105
140	3.3220	0.3603	3.5618	0.3268	3.7544	0.3002
144	3.0580	0.3503	3.2835	0.3226	3.4862	0.2935
148	2.7884	0.3611	2.9972	0.2989	3.1943	0.2607
152	2.5277	0.3232	2.6753	0.3002	2.8322	0.2802

	r/R = 0.737		r/R = 0.603		r/R = 0.470	
θ deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
156	2.1837	0.3144	2.3866	0.2893	2.5039	0.2513
160	1.8452	0.3145	2.0209	0.2851	2.1755	0.2633
164	1.4908	0.2799	1.6438	0.2648	1.7883	0.2344
168	1.1228	0.2775	1.2411	0.2382	1.3799	0.2413
172	0.7508	0.2625	0.8931	0.2269	0.9867	0.2322
176	0.3337	0.2307	0.4792	0.2439	0.5959	0.2159
180	0.0510	0.0739	0.1000	0.1199	0.1430	0.1402

SPRE s/d = 44

	r/R = 0.337		r/R = 0.203		r/R = 0.003	
θ deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
4	0.0034	0.0073	0.0028	0.0034	0.0089	0.0039
8	0.0953	0.0923	0.1665	0.0943	0.0876	0.0472
12	0.3724	0.2215	0.6451	0.1395	0.6456	0.0839
16	1.1067	0.2131	1.1435	0.1744	1.1697	0.1062
20	1.6283	0.1526	1.6145	0.1437	1.6629	0.1404
24	2.0003	0.1190	1.9770	0.1236	2.0012	0.1364
28	2.3484	0.0981	2.3354	0.1065	2.3418	0.1082
32	2.7098	0.0887	2.7001	0.0890	2.7272	0.0909
36	3.0523	0.0783	3.0553	0.0776	3.0967	0.0761
40	3.3502	0.0651	3.3541	0.0636	3.3966	0.0645
44	3.6133	0.0585	3.6275	0.0534	3.6990	0.0614
48	3.9515	0.0504	3.9538	0.0520	4.0242	0.0610
52	4.2540	0.0478	4.2548	0.0425	4.3224	0.0528
56	4.4724	0.0508	4.4778	0.0504	4.5342	0.0497
60	4.6833	0.0500	4.6878	0.0479	4.7682	0.0399
64	4.9177	0.0494	4.9197	0.0463	5.0148	0.0474
68	5.1360	0.0543	5.1507	0.0549	5.2096	0.0450
72	5.2591	0.0318	5.2648	0.0378	5.3599	0.0532
76	5.3855	0.0352	5.4019	0.0423	5.5011	0.0280
80	5.5594	0.0377	5.5734	0.0289	5.6622	0.0357
84	5.6506	0.0407	5.6716	0.0453	5.7843	0.0540
88	5.7263	0.0464	5.7526	0.0322	5.8620	0.0280
92	5.7796	0.0245	5.7825	0.0303	5.8765	0.0195
96	5.8047	0.0783	5.8270	0.0841	5.8934	0.0572
98	5.8067	0.1119	5.8260	0.1126	5.9134	0.0812
100	5.7825	0.1649	5.8144	0.1439	5.9168	0.1237
102	5.7748	0.2305	5.8619	0.2219	5.9570	0.1907
104	5.8543	0.3252	5.9983	0.3054	6.1290	0.2986

	r/R = 0.337		r/R = 0.203		r/R = 0.003	
θ deg.	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec	ũ m/sec	u' m/sec
106	5.8227	0.4172	6.0069	0.3782	6.2211	0.3483
108	5.5471	0.4213	5.6966	0.3846	5.8968	0.3501
110	5.3145	0.3324	5.4396	0.2910	5.5977	0.2781
112	5.1928	0.3056	5.3301	0.2650	5.4856	0.2534
114	5.1612	0.2868	5.2971	0.2533	5.4352	0.2268
116	5.0901	0.2914	5.2512	0.2570	5.3712	0.2329
120	4.9942	0.2883	5.1096	0.2557	5.2327	0.2397
124	4.8021	0.2780	4.9450	0.2556	5.0781	0.2203
128	4.5895	0.2770	4.7309	0.2542	4.8862	0.2298
132	4.3592	0.2813	4.5000	0.2511	4.6287	0.2287
136	4.1404	0.2840	4.2390	0.2449	4.3841	0.2194
140	3.8972	0.2685	3.9911	0.2422	4.1243	0.2161
144	3.5946	0.2557	3.6979	0.2297	3.8392	0.2089
148	3.2920	0.2573	3.3930	0.2227	3.5093	0.1918
152	2.9939	0.2229	3.0697	0.2078	3.1715	0.1985
156	2.6303	0.2503	2.7204	0.2081	2.8216	0.1973
160	2.2829	0.2283	2.3692	0.2045	2.4679	0.1802
164	1.9014	0.2133	1.9774	0.1912	2.0581	0.1838
168	1.4714	0.2067	1.5760	0.1790	1.6365	0.1681
172	1.0937	0.1917	1.1659	0.1772	1.2344	0.1665
176	0.6787	0.1980	0.7642	0.1687	0.8102	0.1605
180	0.1974	0.1482	0.2501	0.1542	0.2926	0.1555

SPRE s/d = 44

θ deg.	u_m m/sec	θ deg.	u_m m/sec	θ deg.	u_m m/sec
2	0.0610	90	4.6306	178	0.2130
4	0.0669	92	4.6340	180	0.0916
6	0.0858	94	4.6470		
8	0.2128	96	4.6503		
10	0.4628	98	4.6403		
12	0.7362	100	4.6238		
14	0.9702	102	4.6077		
16	1.1908	104	4.5708		
18	1.3971	106	4.5633		
20	1.5783	108	4.5133		
22	1.7300	110	4.4450		
24	1.8725	112	4.3779		
26	2.0116	114	4.3304		
28	2.1483	116	4.2695		
30	2.2953	118	4.2076		
32	2.4522	120	4.1425		
34	2.6036	122	4.0606		
36	2.7346	124	3.9881		
38	2.8546	126	3.9148		
40	2.9616	128	3.8149		
42	3.0555	130	3.7078		
44	3.1771	132	3.6081		
46	3.3147	134	3.5120		
48	3.4423	136	3.4102		
50	3.5668	138	3.2966		
52	3.6725	140	3.1850		
54	3.7560	142	3.0609		
56	3.8309	144	2.9274		
58	3.8987	146	2.7944		
60	3.9809	148	2.6590		
62	4.0741	150	2.5110		
64	4.1585	152	2.3685		
66	4.2451	154	2.2159		
68	4.3104	156	2.0763		
70	4.3580	158	1.9201		
72	4.3876	160	1.7551		
74	4.4212	162	1.5980		
76	4.4655	164	1.4134		
78	4.5162	166	1.2270		
80	4.5670	168	1.0551		
82	4.6040	170	0.8909		
84	4.6262	172	0.7215		
86	4.6325	174	0.5478		
88	4.6324	176	0.3773		

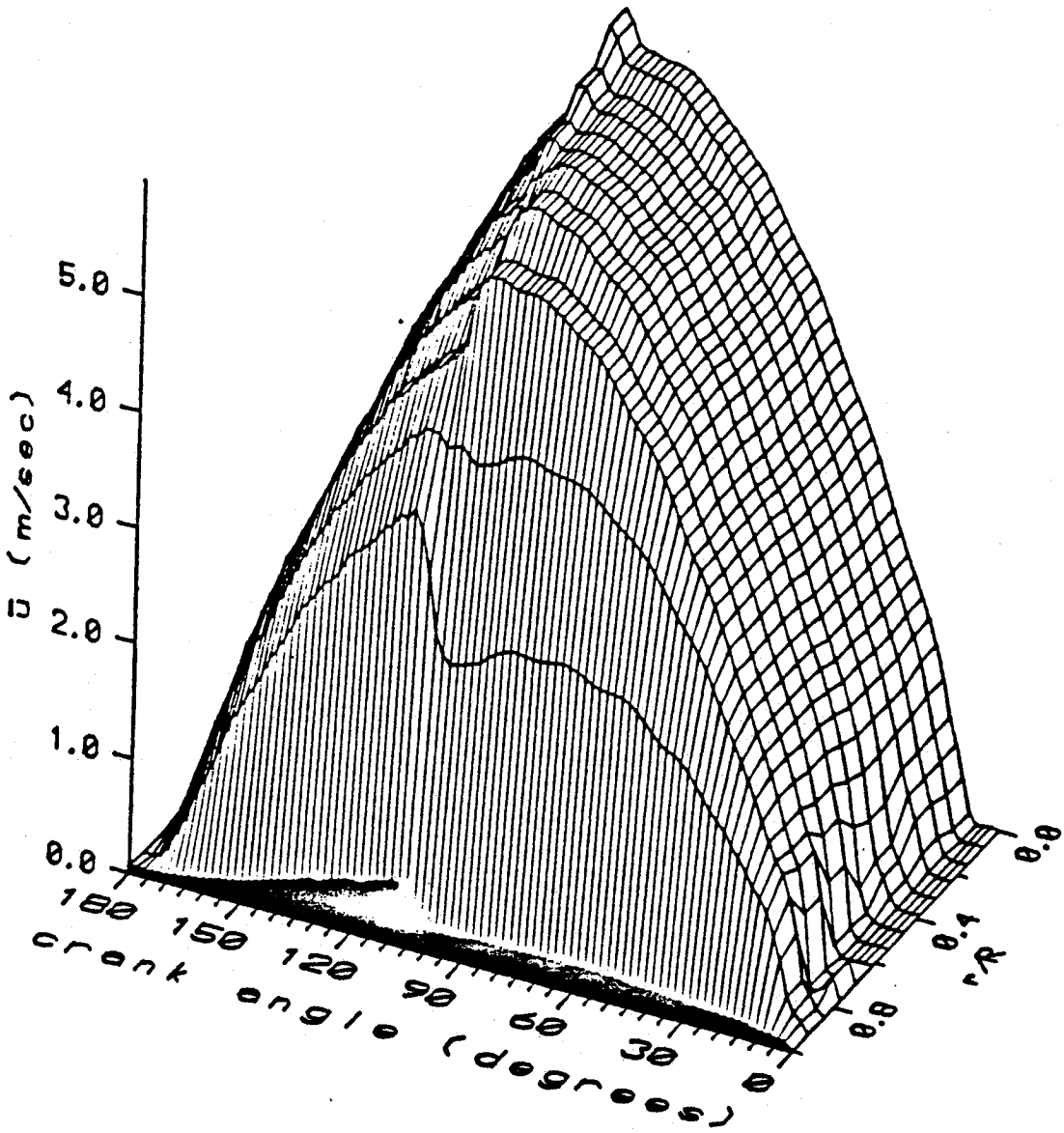


Figure 11: Ensemble-averaged velocity at s/d = 44

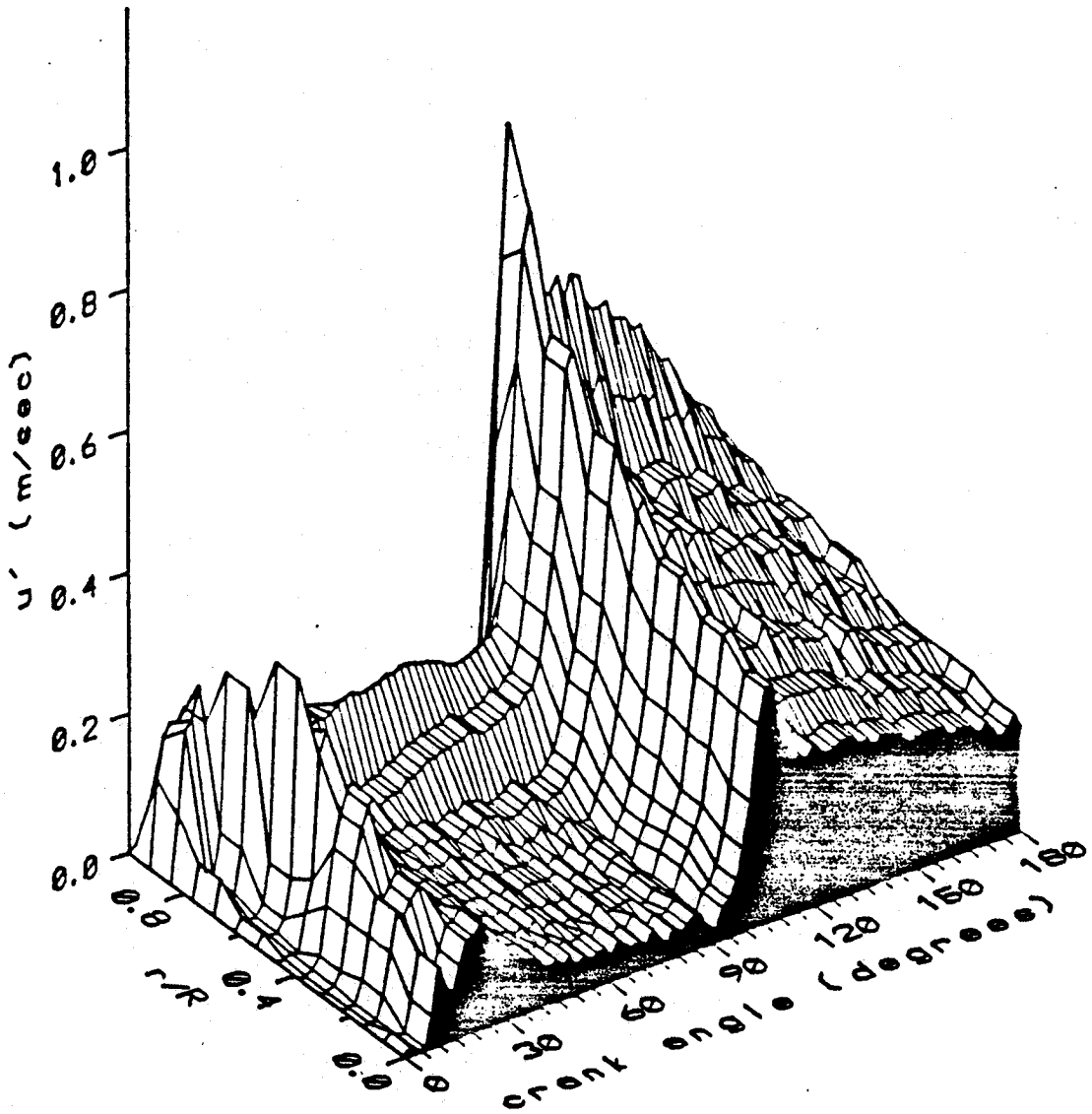


Figure 12: Streamwise velocity fluctuation at s/d = 44

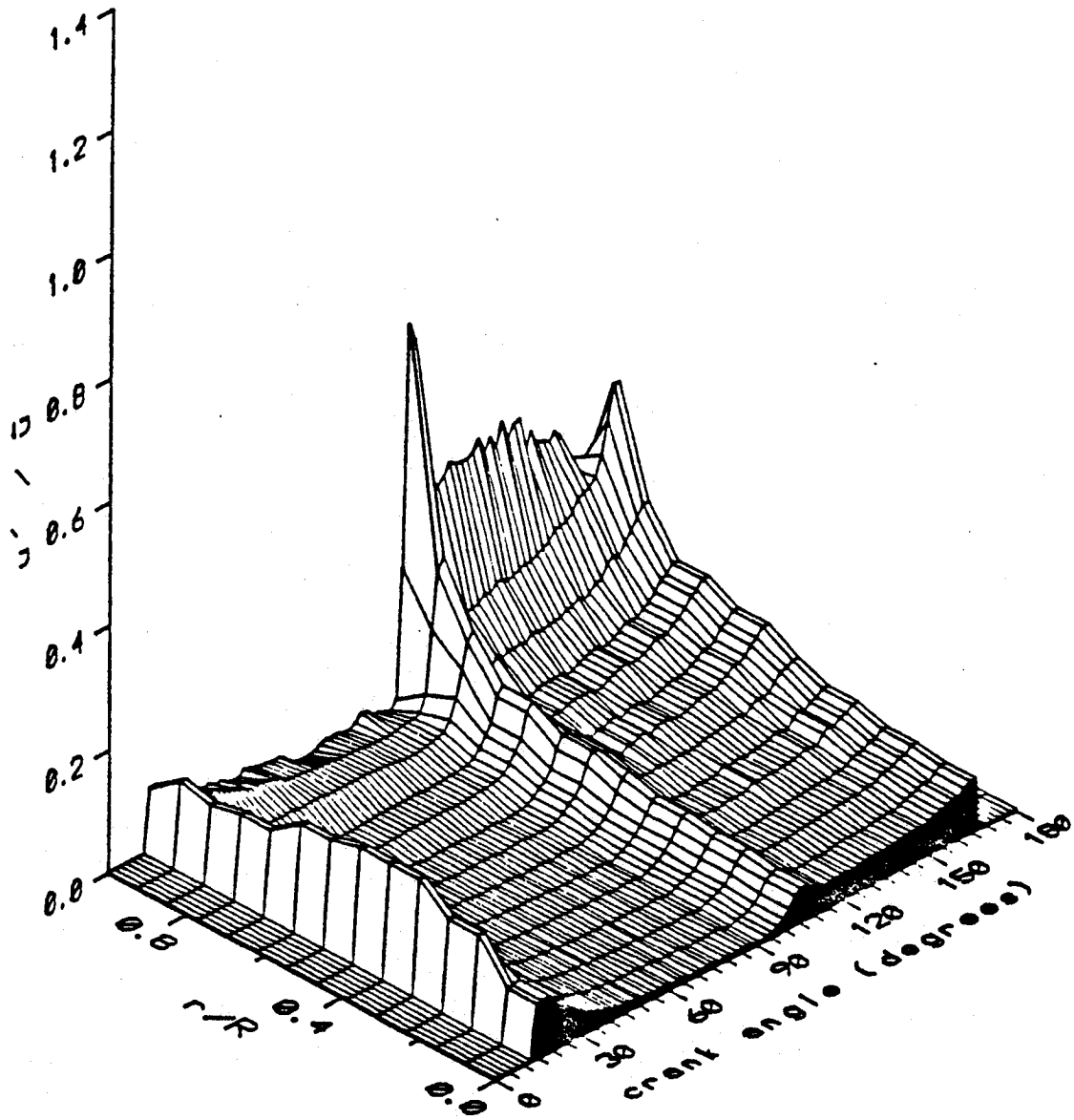


Figure 13: Streamwise turbulence intensity, u' / ū , at s/d = 44

Note: The peaks appear lower than the actual data due to smoothing by the plotting package. The peak turbulence intensity is 1.30, at 164° for r/R = 0.990.

I. CROSS-WIRE DATA

SPRE Operating Point
Nozzle Inlet Geometry

$(\theta, \tilde{u})$ (θ, u') $(\theta, \tilde{v})$ (θ, v') $(\theta, \overline{u'v'})$

Ambient Conditions

axial station s/d	T (^{o}C)	P (bar)
0.33	28.24	0.983
16	27.20	0.986
30	27.45	0.992
44	28.22	0.989

Note that the data are tabulated at every 4 degrees of crank position except within ±10 degrees of transition, for which the resolution is every 2 degrees.

II. SUPPLEMENTAL FIGURES

In order to supplement the three-dimensional figures which appear in the main body of the thesis, additional figures generated from the cross-wire data have been included after the tabulated data for each of the four axial stations. At each station, smoothed plots are provided of the streamwise and radial rms-velocity components, in addition to unsmoothed plots of the Reynolds shear stress.

CROSS-WIRE DATA

SPRE s/d = 0.33

r/R = 0.800

θ	$\tilde{u}$	u'	$\tilde{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.4228	0.1442	-0.0037	0.0510	-0.0004
8	0.7592	0.1309	0.0295	0.0573	-0.0018
12	1.0543	0.1224	0.0476	0.0716	0.0020
14	1.2319	0.1234	0.0660	0.0591	0.0018
16	1.4369	0.1124	0.0754	0.0539	0.0000
18	1.6428	0.1044	0.0936	0.0653	0.0015
20	1.8385	0.1037	0.1082	0.0542	0.0009
22	2.0077	0.1299	0.1301	0.0813	0.0039
24	2.1336	0.1211	0.1404	0.0710	0.0029
26	2.2679	0.1385	0.1355	0.0891	0.0056
28	2.4336	0.1403	0.1627	0.0994	0.0057
30	2.6012	0.1561	0.1706	0.1159	0.0057
32	2.7859	0.1758	0.1716	0.1056	0.0089
36	3.1658	0.1686	0.1972	0.1414	0.0070
40	3.4388	0.2097	0.2112	0.1824	0.0169
44	3.6419	0.2239	0.2313	0.1953	0.0173
48	3.9453	0.2142	0.2472	0.1836	0.0085
52	4.2239	0.1738	0.2153	0.1892	0.0059
56	4.3971	0.1723	0.2403	0.1425	0.0076
60	4.5231	0.1764	0.2434	0.1584	0.0071
64	4.7320	0.1825	0.2496	0.1729	0.0204
68	4.8728	0.1861	0.2542	0.2051	0.0224
72	4.9824	0.1969	0.2824	0.1399	0.0096
76	5.0919	0.1369	0.3099	0.1414	0.0146
80	5.1970	0.1510	0.3071	0.1392	0.0160
84	5.2939	0.1026	0.3170	0.1155	0.0106
88	5.2917	0.0739	0.2946	0.0746	0.0044
92	5.3002	0.1313	0.3098	0.1397	0.0176
96	5.2889	0.0953	0.3048	0.0954	0.0086
100	5.2613	0.0634	0.3048	0.0666	0.0039
104	5.2201	0.1126	0.2949	0.1165	0.0129
108	5.1174	0.0987	0.3022	0.1037	0.0100
112	5.0316	0.1373	0.2963	0.1446	0.0195
116	4.9070	0.1231	0.2786	0.1256	0.0152
120	4.7480	0.1185	0.3032	0.1249	0.0146
124	4.5449	0.1097	0.2526	0.1118	0.0121

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
128	4.3999	0.1006	0.2249	0.1050	0.0105
132	4.1969	0.0728	0.2592	0.0749	0.0047
136	3.9586	0.1116	0.2533	0.1179	0.0130
140	3.6870	0.0887	0.2196	0.0945	0.0083
144	3.4194	0.0704	0.2176	0.0728	0.0050
148	3.1314	0.0618	0.2043	0.0690	0.0041
152	2.7973	0.0831	0.2004	0.0886	0.0072
156	2.4577	0.0473	0.1785	0.0484	0.0022
160	2.1253	0.0461	0.1586	0.0470	0.0021
164	1.7501	0.0232	0.1476	0.0236	0.0004
168	1.3544	0.0451	0.1389	0.0477	0.0021
172	0.9781	0.0345	0.1214	0.0368	0.0012
176	0.5921	0.0295	0.0954	0.0301	0.0009
180	0.2999	0.0232	0.0361	0.0240	0.0006

SPRE s/d = 0.33

r/R = 0.733

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.3664	0.1058	-0.0049	0.0500	-0.0011
8	0.7135	0.1282	0.0232	0.0676	-0.0002
12	1.0306	0.1337	0.0456	0.0724	-0.0009
14	1.2113	0.1191	0.0588	0.0694	-0.0005
16	1.4199	0.1096	0.0736	0.0607	-0.0013
18	1.6272	0.0877	0.0790	0.0596	-0.0001
20	1.8303	0.1073	0.0962	0.0733	0.0012
22	2.0003	0.1187	0.1177	0.0968	0.0029
24	2.1644	0.1338	0.1530	0.0911	0.0024
26	2.3055	0.1386	0.1360	0.1063	0.0039
28	2.4821	0.1385	0.1576	0.1181	0.0049
30	2.6330	0.1660	0.1761	0.1742	0.0155
32	2.8140	0.1590	0.1841	0.1544	0.0078
36	3.1767	0.1732	0.2194	0.1853	0.0134
40	3.4420	0.1616	0.2321	0.1970	0.0020
44	3.6626	0.1792	0.2125	0.2370	0.0109
48	3.9808	0.1829	0.2513	0.2599	0.0106
52	4.2332	0.1763	0.2617	0.2461	0.0106
56	4.3857	0.1564	0.2837	0.2238	0.0095
60	4.5485	0.1330	0.2641	0.2152	0.0127
64	4.7357	0.1910	0.2670	0.2274	0.0231
68	4.9162	0.1487	0.2844	0.1901	0.0150
72	5.0091	0.1193	0.2673	0.1696	0.0128
76	5.0969	0.0775	0.3031	0.1254	0.0059
80	5.2002	0.0469	0.3442	0.0815	0.0009
84	5.2828	0.1057	0.3296	0.1208	0.0105
88	5.2798	0.0716	0.3181	0.0784	0.0050
92	5.2775	0.0628	0.3175	0.0687	0.0039
96	5.2500	0.1074	0.3012	0.1141	0.0119
100	5.2424	0.0846	0.3103	0.0876	0.0073
104	5.1926	0.1193	0.3168	0.1235	0.0143
108	5.0949	0.1313	0.3046	0.1343	0.0173
112	4.9860	0.1141	0.2708	0.1169	0.0131
116	4.8816	0.0756	0.3060	0.0788	0.0056
120	4.7343	0.0743	0.2906	0.0772	0.0056
124	4.5563	0.0957	0.2685	0.0974	0.0092
128	4.4088	0.0346	0.2319	0.0341	0.0011
132	4.1696	0.0568	0.2538	0.0623	0.0034
136	3.9654	0.1157	0.2620	0.1224	0.0141
140	3.6878	0.0928	0.2196	0.0972	0.0090
144	3.4086	0.0670	0.2069	0.0699	0.0045
148	3.1444	0.0700	0.2075	0.0721	0.0048

θ	$\tilde{u}$	u'	$\tilde{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
152	2.8076	0.0660	0.1923	0.0716	0.0046
156	2.4864	0.0684	0.1760	0.0709	0.0047
160	2.1528	0.0452	0.1414	0.0487	0.0022
164	1.7881	0.0485	0.1391	0.0508	0.0024
168	1.4137	0.0461	0.1197	0.0487	0.0021
172	1.0693	0.0461	0.1015	0.0483	0.0022
176	0.6740	0.0433	0.0842	0.0462	0.0020
180	0.3251	0.0183	0.0396	0.0191	0.0004

SPRE s/d = 0.33

r/R = 0.600

θ	$\tilde{u}$	u'	$\tilde{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.3247	0.1047	0.0036	0.0552	-0.0011
8	0.6328	0.1357	0.0164	0.0622	-0.0016
12	0.9604	0.1196	0.0452	0.0771	-0.0003
14	1.1696	0.1264	0.0578	0.0743	-0.0017
16	1.3719	0.1234	0.0733	0.0808	0.0001
18	1.5950	0.1071	0.0856	0.0725	-0.0010
20	1.8077	0.1153	0.1157	0.1029	0.0022
22	1.9792	0.1172	0.1177	0.1146	0.0018
24	2.1155	0.1280	0.1421	0.1207	0.0001
26	2.3120	0.1361	0.1693	0.1488	0.0061
28	2.4598	0.1394	0.1735	0.1568	0.0058
30	2.6511	0.1614	0.1940	0.1955	0.0179
32	2.8592	0.1580	0.1854	0.2201	0.0077
36	3.2352	0.1646	0.2062	0.2200	0.0142
40	3.4552	0.1906	0.2383	0.2542	0.0155
44	3.7004	0.1828	0.2862	0.2898	0.0155
48	3.9645	0.1900	0.2803	0.3293	0.0245
52	4.2599	0.1238	0.2808	0.2867	0.0056
56	4.4090	0.1246	0.3097	0.2657	0.0071
60	4.5489	0.1259	0.2727	0.2558	0.0092
64	4.7591	0.1199	0.2574	0.2644	0.0144
68	4.9289	0.1272	0.2411	0.2222	0.0117
72	5.0243	0.1016	0.2785	0.1822	0.0085
76	5.1025	0.1065	0.2929	0.1606	0.0094
80	5.2054	0.0559	0.3193	0.0976	0.0021
84	5.2799	0.1010	0.3284	0.1151	0.0097
88	5.2733	0.0681	0.3215	0.0772	0.0045

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
92	5.2454	0.0622	0.3134	0.0668	0.0038
96	5.2403	0.0847	0.3212	0.0866	0.0071
100	5.2360	0.1144	0.3349	0.1196	0.0136
104	5.1743	0.1056	0.3301	0.1099	0.0113
108	5.0931	0.0861	0.3355	0.0888	0.0074
112	4.9929	0.0975	0.3098	0.0997	0.0094
116	4.8570	0.0998	0.3093	0.1049	0.0101
120	4.7171	0.0608	0.3012	0.0637	0.0038
124	4.5468	0.0768	0.2864	0.0785	0.0060
128	4.3906	0.0892	0.2402	0.0932	0.0083
132	4.1755	0.0856	0.2858	0.0878	0.0074
136	3.9325	0.0489	0.2534	0.0497	0.0024
140	3.6838	0.0810	0.2389	0.0830	0.0067
144	3.4045	0.0689	0.2253	0.0731	0.0049
148	3.1231	0.0787	0.2017	0.0794	0.0061
152	2.7993	0.0572	0.2005	0.0591	0.0034
156	2.4755	0.0552	0.1762	0.0529	0.0027
160	2.1530	0.0648	0.1570	0.0675	0.0044
164	1.7975	0.0368	0.1485	0.0382	0.0014
168	1.4301	0.0394	0.1197	0.0408	0.0016
172	1.0619	0.0387	0.1018	0.0410	0.0016
176	0.6959	0.0214	0.0721	0.0213	0.0004
180	0.3402	0.0215	0.0377	0.0224	0.0005

SPRE s/d = 0.33

r/R = 0.467

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.3077	0.1003	-0.0018	0.0520	-0.0012
8	0.5992	0.1233	0.0275	0.0735	0.0001
12	0.9130	0.1111	0.0502	0.0765	-0.0011
14	1.1134	0.1174	0.0620	0.0764	-0.0005
16	1.3268	0.1192	0.0898	0.0867	0.0016
18	1.5445	0.1019	0.0884	0.0721	-0.0013
20	1.7668	0.1209	0.1102	0.0940	0.0018
22	1.9459	0.1239	0.1202	0.0929	-0.0014
24	2.1224	0.1452	0.1291	0.1156	-0.0009
26	2.2980	0.1469	0.1561	0.1475	0.0014
28	2.4670	0.1382	0.1779	0.1667	0.0048
30	2.6483	0.1636	0.1920	0.1778	0.0087
32	2.8451	0.1655	0.1974	0.2098	0.0092

θ	$\tilde{u}$	u'	$\tilde{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
36	3.2210	0.1626	0.2130	0.2619	0.0153
40	3.4582	0.1400	0.2215	0.2982	0.0078
44	3.7065	0.1582	0.3072	0.3052	0.0112
48	4.0185	0.1723	0.3064	0.2903	0.0102
52	4.2603	0.1622	0.3171	0.3344	0.0119
56	4.4426	0.1552	0.3230	0.3078	0.0175
60	4.5764	0.1230	0.2668	0.3289	0.0129
64	4.7881	0.1621	0.3165	0.3503	0.0327
68	4.9385	0.1181	0.3041	0.2548	0.0130
72	5.0391	0.1063	0.2746	0.1971	0.0048
76	5.1194	0.1077	0.3098	0.1835	0.0108
80	5.1940	0.1377	0.3236	0.1662	0.0183
84	5.2898	0.0780	0.3435	0.1056	0.0058
88	5.2962	0.0867	0.3494	0.0877	0.0065
92	5.2733	0.0679	0.3534	0.0716	0.0045
96	5.2376	0.0946	0.3301	0.0979	0.0091
100	5.2203	0.0588	0.3405	0.0620	0.0035
104	5.1854	0.1146	0.3419	0.1222	0.0136
108	5.0933	0.0951	0.3489	0.1012	0.0095
112	4.9914	0.0921	0.3297	0.0958	0.0087
116	4.8546	0.0327	0.3130	0.0337	0.0008
120	4.7185	0.0521	0.3294	0.0534	0.0027
124	4.5527	0.0893	0.3168	0.0935	0.0083
128	4.4054	0.0818	0.2813	0.0865	0.0070
132	4.1717	0.0633	0.3045	0.0595	0.0037
136	3.9359	0.1226	0.2787	0.1294	0.0158
140	3.6786	0.0583	0.2557	0.0604	0.0035
144	3.4224	0.1010	0.2630	0.1050	0.0105
148	3.1210	0.0586	0.2140	0.0635	0.0036
152	2.8110	0.0964	0.2299	0.1008	0.0097
156	2.4899	0.0292	0.1982	0.0312	0.0006
160	2.1554	0.0442	0.1741	0.0462	0.0020
164	1.7940	0.0699	0.1576	0.0748	0.0052
168	1.4339	0.0541	0.1309	0.0566	0.0030
172	1.0594	0.0244	0.1013	0.0249	0.0006
176	0.7006	0.0456	0.0748	0.0474	0.0021
180	0.3436	0.0185	0.0390	0.0185	0.0004

144

SPRE s/d = 0.33

r/R = 0.333

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.2765	0.0917	0.0106	0.0495	0.0003
8	0.5441	0.1181	0.0241	0.0652	0.0006
12	0.8879	0.1116	0.0552	0.0724	-0.0015
14	1.0798	0.1082	0.0724	0.0880	0.0020
16	1.2849	0.1122	0.0910	0.0772	-0.0006
18	1.5066	0.1055	0.0964	0.0716	-0.0018
20	1.7443	0.1197	0.1046	0.0998	0.0023
22	1.9078	0.1123	0.1194	0.1014	0.0008
24	2.1076	0.1366	0.1370	0.1370	-0.0001
26	2.2837	0.1514	0.1729	0.1667	0.0065
28	2.4455	0.1836	0.1946	0.2047	0.0143
30	2.6364	0.1511	0.1799	0.1980	0.0052
32	2.8121	0.1670	0.2002	0.2111	0.0010
36	3.2031	0.1723	0.2196	0.2630	0.0034
40	3.4561	0.1590	0.2212	0.2825	0.0058
44	3.7039	0.1710	0.2452	0.2973	0.0051
48	4.0023	0.1548	0.2659	0.3514	0.0043
52	4.2537	0.1353	0.2960	0.3183	-0.0010
56	4.4219	0.1468	0.3265	0.3456	0.0136
60	4.5761	0.1266	0.3452	0.3380	0.0071
64	4.7592	0.1043	0.2448	0.3461	0.0026
68	4.9344	0.1148	0.2454	0.2656	0.0054
72	5.0235	0.1139	0.2724	0.2250	0.0122
76	5.0866	0.1018	0.2967	0.1809	0.0101
80	5.1816	0.0859	0.3311	0.1231	0.0071
84	5.2461	0.0657	0.3400	0.0835	0.0046
88	5.2542	0.0943	0.3444	0.1024	0.0094
92	5.2485	0.1339	0.3463	0.1413	0.0188
96	5.2397	0.0399	0.3396	0.0417	0.0017
100	5.2159	0.1483	0.3329	0.1559	0.0228
104	5.1651	0.1175	0.3584	0.1223	0.0142
108	5.0851	0.0913	0.3520	0.0956	0.0087
112	4.9681	0.0902	0.3509	0.0939	0.0083
116	4.8316	0.1136	0.3394	0.1185	0.0134
120	4.7105	0.0858	0.3297	0.0943	0.0080
124	4.5350	0.0930	0.3312	0.0984	0.0089
128	4.4106	0.0874	0.2623	0.0916	0.0080
132	4.1578	0.0475	0.3016	0.0489	0.0023
136	3.9079	0.0909	0.2683	0.0931	0.0082
140	3.6807	0.0714	0.2474	0.0716	0.0050
144	3.3966	0.0563	0.2417	0.0587	0.0033
148	3.1047	0.0627	0.2283	0.0648	0.0039
152	2.7924	0.0412	0.2179	0.0431	0.0017

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
156	2.4764	0.0353	0.1981	0.0367	0.0012
160	2.1530	0.0246	0.1623	0.0237	0.0005
164	1.7885	0.0499	0.1531	0.0527	0.0026
168	1.4209	0.0369	0.1188	0.0380	0.0013
172	1.0622	0.0436	0.1024	0.0457	0.0020
176	0.6963	0.0184	0.0705	0.0186	0.0003
180	0.3390	0.0177	0.0333	0.0181	0.0003

SPRE s/d = 0.33

r/R = 0.200

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.2680	0.0819	0.0239	0.0517	0.0013
8	0.5104	0.1170	0.0326	0.0698	0.0029
12	0.8276	0.1070	0.0584	0.0719	0.0012
14	1.0172	0.1009	0.0570	0.0707	-0.0001
16	1.2379	0.1052	0.0929	0.1064	0.0042
18	1.4832	0.1082	0.1036	0.0944	0.0017
20	1.6854	0.1141	0.1082	0.0875	0.0008
22	1.9022	0.1199	0.1217	0.1123	0.0008
24	2.1114	0.1404	0.1246	0.1522	0.0000
26	2.2640	0.1750	0.1690	0.1858	0.0054
28	2.4391	0.1743	0.1794	0.2013	0.0081
30	2.5989	0.1865	0.2095	0.2315	0.0128
32	2.7805	0.1969	0.2058	0.2629	0.0104
36	3.2043	0.1662	0.2020	0.2893	0.0150
40	3.4770	0.1621	0.2499	0.3464	0.0038
44	3.7060	0.1558	0.2290	0.3558	0.0104
48	3.9969	0.1696	0.2845	0.3588	0.0108
52	4.2757	0.1491	0.3202	0.3576	0.0084
56	4.4136	0.1386	0.2454	0.3219	0.0023
60	4.5646	0.1172	0.2317	0.3103	0.0078
64	4.7613	0.1329	0.2139	0.3182	0.0164
68	4.9276	0.1133	0.2564	0.2924	0.0095
72	5.0105	0.1192	0.2538	0.2613	0.0142
76	5.0777	0.0543	0.2921	0.1591	0.0009
80	5.1797	0.1556	0.3236	0.1873	0.0259
84	5.2436	0.0804	0.3327	0.0973	0.0063
88	5.2331	0.1066	0.3237	0.1112	0.0114
92	5.2386	0.1216	0.3433	0.1263	0.0152
96	5.2107	0.1115	0.3168	0.1150	0.0127
100	5.2147	0.0769	0.3287	0.0802	0.0060

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
104	5.1763	0.1182	0.3744	0.1231	0.0143
108	5.0511	0.0952	0.3239	0.0984	0.0093
112	4.9579	0.0600	0.3419	0.0590	0.0032
116	4.8284	0.0595	0.3376	0.0626	0.0036
120	4.7044	0.0704	0.3185	0.0735	0.0049
124	4.5092	0.0844	0.2968	0.0842	0.0067
128	4.3919	0.0810	0.2494	0.0833	0.0067
132	4.1503	0.0681	0.2926	0.0687	0.0045
136	3.9165	0.0709	0.2659	0.0759	0.0051
140	3.6631	0.0726	0.2254	0.0765	0.0055
144	3.3889	0.0626	0.2377	0.0660	0.0041
148	3.0958	0.0723	0.2089	0.0742	0.0052
152	2.7793	0.0708	0.1977	0.0733	0.0051
156	2.4644	0.0728	0.1863	0.0755	0.0054
160	2.1457	0.0629	0.1568	0.0655	0.0041
164	1.7800	0.0312	0.1478	0.0314	0.0009
168	1.4234	0.0272	0.1225	0.0276	0.0007
172	1.0645	0.0486	0.1069	0.0514	0.0025
176	0.6995	0.0346	0.0737	0.0359	0.0012
180	0.3433	0.0193	0.0338	0.0194	0.0004

SPRE s/d = 0.33

r/R = 0.000

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.2841	0.0974	0.0335	0.0692	0.0043
8	0.5308	0.1203	0.0391	0.0798	0.0043
12	0.8414	0.1055	0.0658	0.0767	0.0037
14	1.0189	0.1084	0.0852	0.0864	0.0035
16	1.2365	0.0996	0.0919	0.0806	0.0040
18	1.4489	0.1105	0.1012	0.0770	0.0035
20	1.6766	0.1064	0.1022	0.0862	0.0035
22	1.8678	0.1090	0.1053	0.1025	0.0031
24	2.0961	0.1304	0.1251	0.1381	0.0004
26	2.2359	0.1449	0.1226	0.1762	0.0036
28	2.4317	0.1690	0.1519	0.1902	-0.0002
30	2.6295	0.1640	0.1938	0.2010	0.0009
32	2.7983	0.1758	0.2081	0.2459	0.0020
36	3.1809	0.1436	0.1836	0.2684	0.0023
40	3.4778	0.1585	0.1434	0.2874	0.0091
44	3.6992	0.1635	0.1214	0.3242	0.0065

θ	$\tilde{u}$	u'	$\tilde{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
48	3.9608	0.1503	0.2113	0.3488	-0.0012
52	4.2308	0.1265	0.2278	0.3350	0.0053
56	4.3925	0.1165	0.2230	0.3806	0.0051
60	4.5378	0.1372	0.2474	0.3732	0.0034
64	4.7321	0.1326	0.2243	0.3531	0.0166
68	4.9011	0.1260	0.2555	0.3105	0.0116
72	4.9761	0.1098	0.2505	0.2421	0.0055
76	5.0468	0.1116	0.2889	0.1859	0.0117
80	5.1526	0.1415	0.3305	0.1724	0.0214
84	5.2159	0.1005	0.3361	0.1144	0.0101
88	5.2065	0.0476	0.3263	0.0514	0.0021
92	5.2049	0.0795	0.3326	0.0850	0.0067
96	5.2008	0.1117	0.3295	0.1168	0.0130
100	5.1804	0.0834	0.3267	0.0855	0.0068
104	5.1319	0.1195	0.3558	0.1260	0.0148
108	5.0413	0.0761	0.3348	0.0796	0.0060
112	4.9076	0.0972	0.3144	0.1007	0.0096
116	4.8138	0.1060	0.3414	0.1127	0.0118
120	4.6694	0.0995	0.3037	0.1066	0.0104
124	4.4744	0.0939	0.2822	0.0980	0.0088
128	4.3596	0.0675	0.2348	0.0709	0.0048
132	4.1212	0.0949	0.2770	0.0985	0.0092
136	3.8993	0.0697	0.2623	0.0688	0.0046
140	3.6543	0.0839	0.2326	0.0876	0.0073
144	3.3636	0.0593	0.2246	0.0612	0.0036
148	3.0873	0.0500	0.2112	0.0551	0.0025
152	2.7623	0.0634	0.1914	0.0654	0.0040
156	2.4553	0.0926	0.1848	0.0982	0.0090
160	2.1314	0.0268	0.1512	0.0274	0.0007
164	1.7689	0.0514	0.1400	0.0526	0.0026
168	1.4200	0.0473	0.1219	0.0491	0.0023
172	1.0522	0.0233	0.0976	0.0231	0.0005
176	0.6964	0.0342	0.0703	0.0358	0.0012
180	0.3429	0.0196	0.0348	0.0198	0.0004

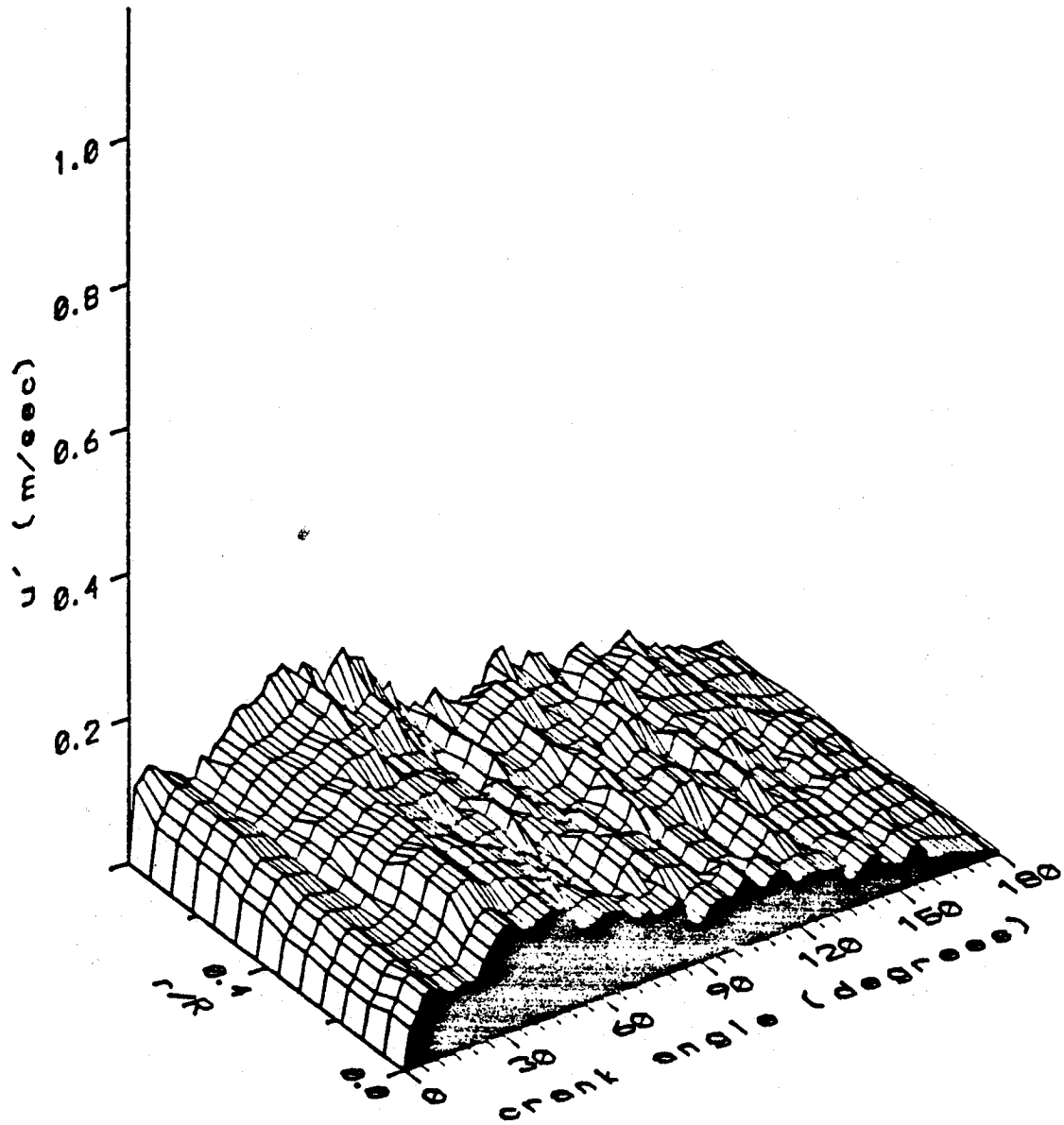


Figure 14: Streamwise velocity fluctuation at s/d = 0.33 (smoothed)

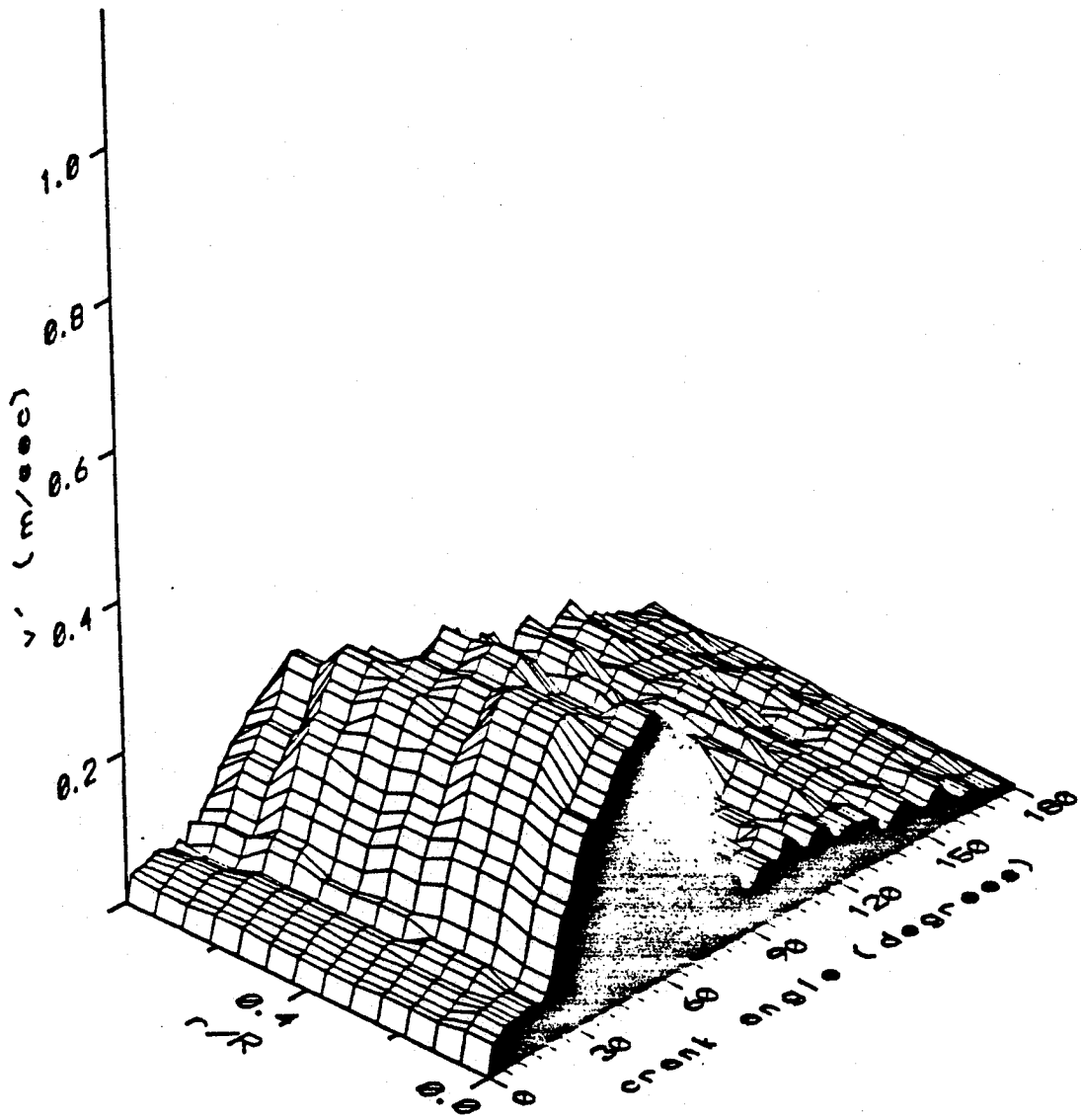


Figure 15: Radial velocity fluctuation at s/d = 0.33 (smoothed)

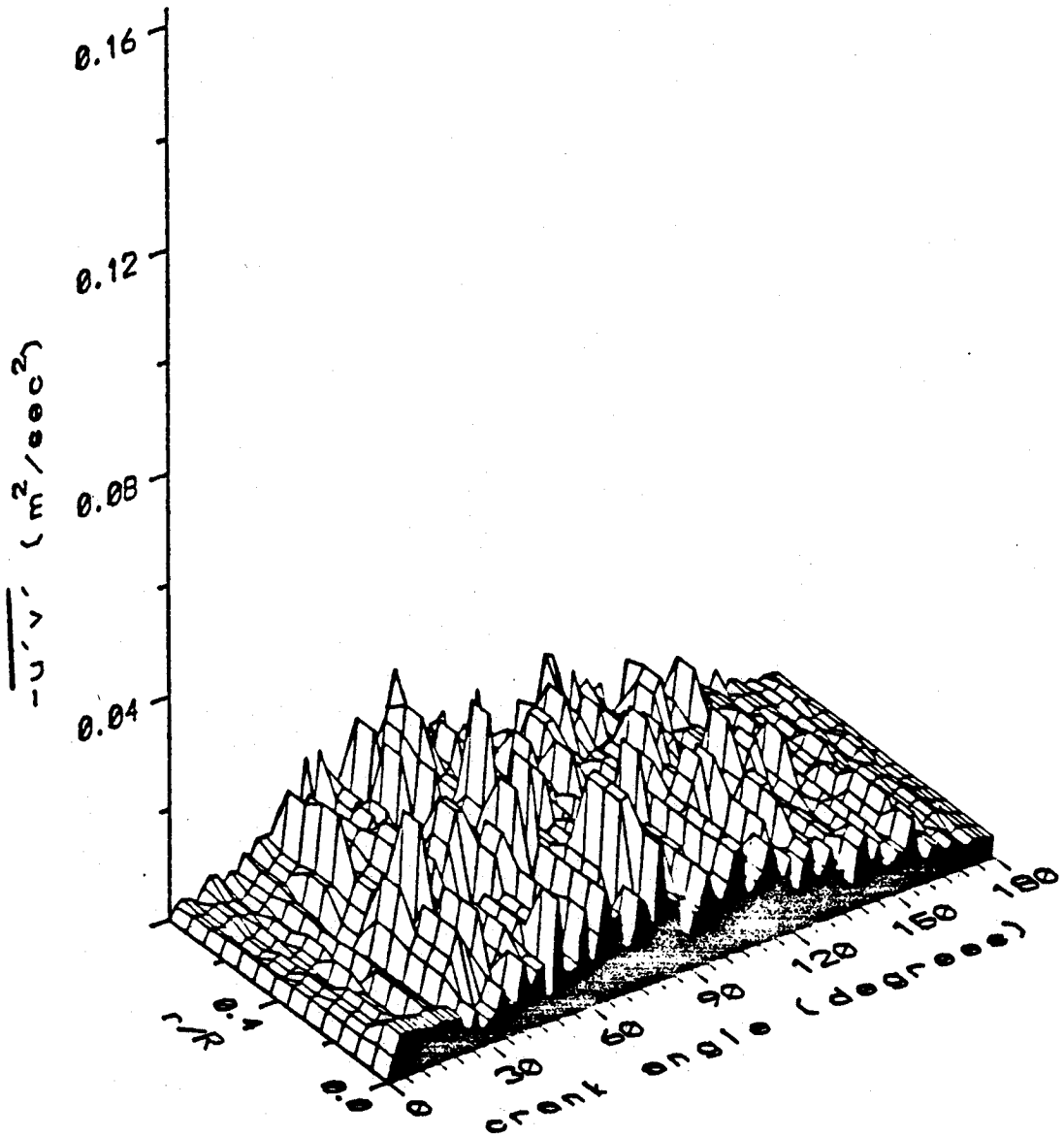


Figure 16 : Reynolds shear stress at s/d = 0.33
(not smoothed)

CROSS-WIRE DATA

SPRE s/d = 16

r/R = 0.800

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.4461	0.1531	-0.0060	0.0566	0.0010
8	0.8037	0.1336	0.0199	0.0735	-0.0004
12	1.1030	0.1234	0.0571	0.0703	0.0006
16	1.5007	0.1087	0.0841	0.0581	0.0010
20	1.9468	0.1019	0.1167	0.0631	0.0024
24	2.2698	0.0968	0.1360	0.0679	0.0035
28	2.5781	0.0911	0.1770	0.0623	0.0028
32	2.9333	0.1199	0.1995	0.0922	0.0067
36	3.3227	0.1164	0.2305	0.0816	0.0064
40	3.5714	0.1280	0.2396	0.1034	0.0089
44	3.8202	0.1411	0.2781	0.0829	0.0074
48	4.1732	0.1635	0.2782	0.1156	0.0067
52	4.4142	0.1395	0.3173	0.0534	0.0034
54	4.5015	0.1671	0.3336	0.0944	0.0096
56	4.5817	0.1898	0.3419	0.1083	0.0139
58	4.6959	0.2362	0.3528	0.1052	0.0124
60	4.7340	0.3192	0.3514	0.1118	0.0153
62	4.8223	0.3949	0.3431	0.1590	0.0369
64	4.9051	0.4097	0.3611	0.1708	0.0382
66	4.9671	0.4536	0.3851	0.2046	0.0291
68	5.0518	0.5151	0.3246	0.2530	0.0567
70	5.1391	0.5232	0.3634	0.2312	0.0421
72	5.1344	0.5465	0.3595	0.2293	0.0599
76	5.2278	0.5225	0.3398	0.2256	0.0460
80	5.3008	0.5411	0.3370	0.2999	0.0743
84	5.2597	0.5891	0.3349	0.2936	0.0638
88	5.2434	0.5874	0.3493	0.2908	0.0559
92	5.2642	0.6368	0.4193	0.3119	0.1043
96	5.2715	0.5566	0.3542	0.2667	0.0471
100	5.2955	0.5882	0.3632	0.2820	0.0786
104	5.3362	0.4626	0.3934	0.1723	0.0142
108	5.2548	0.4841	0.4154	0.1125	0.0118
112	5.1616	0.3411	0.4266	0.1247	0.0085
116	4.9727	0.2973	0.4407	0.0902	0.0128
120	4.7500	0.2439	0.4476	0.1410	0.0203
124	4.5686	0.1599	0.4494	0.1320	0.0166

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
128	4.3586	0.1254	0.4086	0.0663	-0.0009
132	4.1215	0.1243	0.4347	0.1115	0.0107
136	3.9179	0.1056	0.4380	0.0920	0.0079
140	3.6605	0.1034	0.4228	0.0951	0.0088
144	3.3768	0.0920	0.3932	0.0922	0.0074
148	3.0425	0.0626	0.3779	0.0568	0.0028
152	2.7014	0.0435	0.3563	0.0389	0.0013
156	2.3705	0.0386	0.3277	0.0338	0.0010
160	2.0101	0.0624	0.3065	0.0644	0.0038
164	1.6036	0.0573	0.2730	0.0576	0.0031
168	1.2065	0.0452	0.2402	0.0423	0.0017
172	0.8648	0.0336	0.2008	0.0310	0.0008
176	0.5390	0.0362	0.1612	0.0345	0.0011
180	0.3578	0.0289	0.0410	0.0276	0.0007

SPRE s/d = 16

r/R = 0.733

θ	$\tilde{u}$	u'	$\tilde{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.3959	0.1283	-0.0041	0.0563	-0.0025
8	0.7566	0.1304	0.0217	0.0717	0.0002
12	1.0760	0.1157	0.0432	0.0656	-0.0006
16	1.4949	0.1023	0.0901	0.0658	0.0004
20	1.9415	0.1146	0.1185	0.0858	0.0044
24	2.2792	0.1143	0.1475	0.0798	0.0033
28	2.6022	0.1089	0.1761	0.0870	0.0054
32	2.9995	0.0850	0.1976	0.0555	0.0019
36	3.3719	0.1112	0.2143	0.0923	0.0071
40	3.6741	0.1123	0.2460	0.0792	0.0049
44	3.9309	0.1057	0.2643	0.0763	0.0053
48	4.3261	0.1448	0.2489	0.1288	0.0139
52	4.5861	0.1121	0.3184	0.0866	0.0062
54	4.7064	0.1205	0.3323	0.0868	0.0066
56	4.8065	0.1855	0.3396	0.1369	0.0174
58	4.8828	0.1976	0.3443	0.1036	0.0110
60	4.9888	0.2962	0.3348	0.1631	0.0271
62	5.1037	0.3097	0.3751	0.1806	0.0317
64	5.1977	0.3501	0.3478	0.1888	0.0342
66	5.2748	0.3678	0.3749	0.2066	0.0231
68	5.2406	0.4041	0.3542	0.2247	0.0342
70	5.2651	0.4778	0.3629	0.2332	0.0577
72	5.2088	0.5003	0.3387	0.2055	0.0491
76	5.2243	0.5191	0.3378	0.2562	0.0584
80	5.4144	0.4708	0.3556	0.2745	0.0557
84	5.4859	0.4908	0.3739	0.2352	0.0378
88	5.4460	0.4994	0.3514	0.2629	0.0423
92	5.4537	0.5293	0.3849	0.2439	0.0360
96	5.5560	0.5096	0.4061	0.2545	0.0523
100	5.5789	0.4863	0.3848	0.2151	0.0390
104	5.6071	0.4080	0.3945	0.1376	0.0076
108	5.6085	0.2896	0.4071	0.1152	0.0005
112	5.5781	0.2202	0.3944	0.0800	0.0035
116	5.4739	0.1394	0.4249	0.0739	0.0007
120	5.3386	0.1402	0.4785	0.0953	0.0085
124	5.0791	0.1073	0.4222	0.0889	0.0071
128	4.8739	0.1324	0.4280	0.1256	0.0150
132	4.6500	0.0723	0.4170	0.0742	0.0047
136	4.4494	0.1110	0.3380	0.1129	0.0123
140	4.1362	0.0692	0.3517	0.0667	0.0041
144	3.8373	0.1176	0.3399	0.1194	0.0134
148	3.4866	0.0781	0.3376	0.0766	0.0056

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
152	3.1337	0.0601	0.3227	0.0606	0.0034
156	2.7725	0.0665	0.2946	0.0659	0.0041
160	2.4114	0.0701	0.2676	0.0720	0.0047
164	1.9758	0.0430	0.2533	0.0394	0.0015
168	1.5457	0.0289	0.2198	0.0305	0.0008
172	1.1652	0.0541	0.2110	0.0554	0.0029
176	0.7725	0.0366	0.1651	0.0361	0.0011
180	0.4245	0.0338	0.0346	0.0247	0.0001

SPRE s/d = 16

r/R = 0.600

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.3400	0.1171	-0.0036	0.0496	-0.0012
8	0.6646	0.1249	0.0194	0.0594	-0.0017
12	1.0010	0.1302	0.0536	0.0869	0.0004
16	1.4467	0.1179	0.0915	0.0746	-0.0002
20	1.8810	0.1041	0.1147	0.0826	-0.0008
24	2.2134	0.1046	0.1348	0.0792	0.0009
28	2.5658	0.0970	0.1677	0.0542	-0.0005
32	2.9756	0.0890	0.2065	0.0608	-0.0004
36	3.3651	0.1069	0.2126	0.0833	0.0033
40	3.6937	0.1330	0.2560	0.1111	0.0108
44	3.9772	0.0982	0.2723	0.0807	0.0040
48	4.3728	0.1219	0.2943	0.1234	0.0144
52	4.6290	0.1120	0.3089	0.0993	0.0091
54	4.7919	0.1232	0.3464	0.1157	0.0103
56	4.8915	0.1208	0.3311	0.0926	0.0060
58	5.0003	0.1331	0.3446	0.0998	0.0042
60	5.0568	0.2082	0.3500	0.1754	0.0186
62	5.1864	0.2008	0.3668	0.1688	0.0174
64	5.2344	0.2868	0.3655	0.1961	0.0232
66	5.3282	0.3234	0.3805	0.2483	0.0505
68	5.4059	0.2559	0.3693	0.1855	0.0175
70	5.4157	0.2882	0.3807	0.2125	0.0205
72	5.4212	0.3825	0.3678	0.2563	0.0512
76	5.4886	0.3850	0.3822	0.2088	0.0389
80	5.5916	0.3749	0.3998	0.2339	0.0352

θ	$\tilde{u}$	u'	$\tilde{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
84	5.6439	0.4117	0.3894	0.2156	0.0374
88	5.6832	0.3648	0.3911	0.2612	0.0480
92	5.6803	0.3998	0.3812	0.2433	0.0500
96	5.7514	0.3041	0.4095	0.1740	0.0067
100	5.7519	0.2929	0.3704	0.1548	0.0145
104	5.7867	0.2282	0.3991	0.1268	0.0037
108	5.7620	0.1755	0.3578	0.1341	0.0149
112	5.7745	0.1600	0.4162	0.1602	0.0235
116	5.6741	0.0990	0.3732	0.0984	0.0094
120	5.5116	0.1268	0.3575	0.1291	0.0159
124	5.3976	0.1307	0.4243	0.1384	0.0177
128	5.1322	0.1089	0.3580	0.1132	0.0118
132	4.9232	0.1176	0.3531	0.1254	0.0141
136	4.6792	0.1035	0.3493	0.1059	0.0108
140	4.4060	0.1118	0.3372	0.1191	0.0132
144	4.1160	0.0957	0.3005	0.1001	0.0094
148	3.7994	0.0377	0.2928	0.0390	0.0014
152	3.4428	0.0473	0.2507	0.0473	0.0021
156	3.1044	0.0683	0.2279	0.0717	0.0048
160	2.7371	0.0540	0.2195	0.0551	0.0029
164	2.3431	0.0878	0.2058	0.0925	0.0081
168	1.9013	0.0540	0.1665	0.0556	0.0029
172	1.4979	0.0537	0.1509	0.0561	0.0029
176	1.0748	0.0326	0.1280	0.0326	0.0010
180	0.6170	0.0308	0.0881	0.0288	0.0008

SPRE s/d = 16

r/R = 0.467

θ	$\tilde{u}$	u'	$\tilde{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.2885	0.0957	0.0072	0.0489	-0.0008
8	0.5994	0.1157	0.0338	0.0626	0.0002
12	0.9171	0.1223	0.0569	0.0775	-0.0007
16	1.3598	0.1152	0.1032	0.0891	0.0014
20	1.8238	0.1177	0.1302	0.0893	0.0025
24	2.1564	0.1150	0.1470	0.0886	0.0033
28	2.4953	0.1001	0.1667	0.0765	0.0002
32	2.9165	0.1161	0.1948	0.0973	0.0050
36	3.3663	0.1207	0.2489	0.1232	0.0093

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
40	3.6422	0.1403	0.2516	0.1371	0.0153
44	3.9454	0.1039	0.2574	0.0857	0.0011
48	4.2960	0.1027	0.2976	0.1053	0.0094
52	4.6229	0.1026	0.3180	0.0929	0.0066
54	4.7390	0.0770	0.3285	0.0736	0.0019
56	4.8534	0.0981	0.3407	0.0982	0.0044
58	4.9455	0.1462	0.3216	0.1585	0.0133
60	5.0990	0.1772	0.3844	0.1667	0.0151
62	5.1822	0.1672	0.3863	0.1495	0.0067
64	5.2697	0.2324	0.3676	0.1824	0.0180
66	5.3522	0.2532	0.3829	0.2054	0.0241
68	5.4424	0.2411	0.3705	0.2310	0.0219
70	5.4779	0.2374	0.3850	0.1814	0.0082
72	5.5005	0.2533	0.3856	0.2211	0.0269
76	5.5313	0.2945	0.3516	0.2077	0.0270
80	5.6663	0.2809	0.4191	0.2047	0.0203
84	5.6917	0.3367	0.3728	0.2328	0.0367
88	5.7297	0.2844	0.3752	0.2056	0.0249
92	5.7733	0.2995	0.4075	0.2558	0.0396
96	5.8177	0.2320	0.4170	0.1552	0.0147
100	5.8076	0.2086	0.3768	0.1602	0.0190
104	5.8230	0.1642	0.4038	0.1662	0.0203
108	5.8025	0.1210	0.4168	0.1081	0.0106
112	5.7447	0.1433	0.4037	0.1449	0.0201
116	5.6198	0.0297	0.3886	0.0273	0.0003
120	5.4652	0.0963	0.3625	0.0965	0.0086
124	5.2933	0.0467	0.3903	0.0471	0.0017
128	5.1837	0.0757	0.4545	0.0768	0.0052
132	4.9049	0.1421	0.3785	0.1476	0.0207
136	4.6350	0.1102	0.3279	0.1133	0.0121
140	4.4015	0.1054	0.3368	0.1103	0.0115
144	4.1696	0.1079	0.2099	0.1113	0.0119
148	3.7784	0.1026	0.2675	0.1051	0.0106
152	3.4460	0.1026	0.2522	0.1039	0.0105
156	3.1197	0.0479	0.2055	0.0487	0.0022
160	2.7527	0.0665	0.2009	0.0689	0.0045
164	2.3657	0.0688	0.1702	0.0673	0.0043
168	1.9424	0.0593	0.1360	0.0638	0.0037
172	1.5651	0.0179	0.1150	0.0164	0.0002
176	1.1662	0.0386	0.0929	0.0404	0.0015
180	0.7099	0.0295	0.0612	0.0245	0.0006

SPRE s/d = 16

r/R = 0.333

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.2750	0.0749	0.0082	0.0514	0.0000
8	0.5532	0.1104	0.0305	0.0647	0.0001
12	0.8841	0.1245	0.0522	0.0847	0.0008
16	1.3282	0.1140	0.0939	0.0843	0.0014
20	1.7686	0.1119	0.1225	0.0948	0.0025
24	2.1348	0.0989	0.1459	0.0738	-0.0004
28	2.4811	0.1038	0.1674	0.0960	0.0033
32	2.8873	0.0888	0.2021	0.0732	0.0007
36	3.3309	0.1192	0.2316	0.0979	0.0071
40	3.6259	0.0932	0.2355	0.0764	0.0030
44	3.9425	0.0897	0.2615	0.0799	0.0032
48	4.3287	0.0784	0.2786	0.0796	0.0055
52	4.6301	0.0978	0.3197	0.0923	0.0062
54	4.7453	0.0930	0.3278	0.0928	0.0043
56	4.8557	0.1103	0.3403	0.1130	0.0056
58	4.9339	0.1442	0.3333	0.1477	0.0094
60	5.0646	0.1653	0.3273	0.1608	0.0123
62	5.2060	0.1962	0.3843	0.2204	0.0233
64	5.3117	0.1798	0.3647	0.1888	0.0085
66	5.3953	0.1887	0.3491	0.2004	0.0111
68	5.4580	0.2062	0.3878	0.2018	0.0101
70	5.5044	0.1760	0.3942	0.2006	0.0078
72	5.5085	0.1990	0.3518	0.1783	0.0049
76	5.5685	0.2130	0.3626	0.2173	0.0061
80	5.6428	0.2533	0.3763	0.2336	0.0300
84	5.7693	0.2253	0.3886	0.2200	0.0163
88	5.7677	0.2515	0.3738	0.2142	0.0187
92	5.7785	0.2693	0.3636	0.2086	0.0316
96	5.8558	0.2114	0.3917	0.1597	0.0136
100	5.8652	0.1518	0.3908	0.1278	0.0080
104	5.8212	0.1086	0.3865	0.1003	0.0076
108	5.7888	0.1076	0.3819	0.1143	0.0103
112	5.7251	0.0812	0.3824	0.0789	0.0058
116	5.6333	0.0655	0.3566	0.0683	0.0043
120	5.4829	0.0520	0.3424	0.0551	0.0026
124	5.3853	0.1119	0.4304	0.1167	0.0128
128	5.1120	0.1015	0.3485	0.1075	0.0107
132	4.9100	0.0834	0.3385	0.0873	0.0072
136	4.6669	0.0896	0.3477	0.0906	0.0079
140	4.3914	0.0956	0.3142	0.0988	0.0093
144	4.1793	0.0961	0.2357	0.0981	0.0044

θ	$\tilde{u}$	u'	$\tilde{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
148	3.8075	0.1122	0.2753	0.1157	0.0128
152	3.4559	0.1035	0.2391	0.1076	0.0111
156	3.1281	0.0910	0.2075	0.0950	0.0085
160	2.7745	0.0512	0.1933	0.0520	0.0025
164	2.3627	0.0593	0.1544	0.0602	0.0035
168	1.9742	0.0354	0.1433	0.0372	0.0013
172	1.5846	0.0572	0.1146	0.0585	0.0033
176	1.1839	0.0505	0.0825	0.0520	0.0026
180	0.7397	0.0295	0.0525	0.0280	0.0007

SPRE s/d = 16

r/R = 0.200

θ	$\tilde{u}$	u'	$\tilde{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.2581	0.0747	0.0135	0.0502	0.0012
8	0.5406	0.1208	0.0423	0.0796	0.0037
12	0.8655	0.1202	0.0585	0.0855	0.0027
16	1.3002	0.0964	0.1156	0.0841	0.0016
20	1.7517	0.1178	0.1306	0.0903	0.0036
24	2.0978	0.1142	0.1355	0.0889	0.0038
28	2.4436	0.1025	0.1646	0.0990	0.0039
32	2.8839	0.1040	0.1997	0.0807	0.0018
36	3.3141	0.1152	0.2245	0.1024	0.0074
40	3.6296	0.1025	0.2461	0.0925	0.0049
44	3.9391	0.0955	0.2702	0.0856	0.0043
48	4.3095	0.1089	0.2791	0.1111	0.0116
52	4.6123	0.1164	0.3193	0.1215	0.0121
54	4.7259	0.0939	0.3230	0.0891	0.0039
56	4.8315	0.1119	0.3143	0.1308	0.0072
58	4.9372	0.1316	0.3314	0.1386	0.0066
60	5.0475	0.1884	0.3403	0.1829	0.0138
62	5.1949	0.1613	0.3543	0.1851	0.0076
64	5.3256	0.1582	0.3627	0.1917	0.0092
66	5.4059	0.1674	0.3647	0.2020	0.0019
68	5.4983	0.2243	0.3909	0.2427	0.0268
70	5.4850	0.1975	0.4156	0.2045	-0.0005
72	5.5086	0.2057	0.3728	0.2289	0.0128
76	5.6063	0.2362	0.3818	0.2366	0.0132
80	5.6385	0.2452	0.3680	0.2232	0.0146

θ	$\tilde{u}$	u'	$\tilde{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
84	5.7790	0.1969	0.3778	0.2070	0.0051
88	5.8187	0.2362	0.3852	0.2380	0.0210
92	5.8154	0.2267	0.3881	0.1988	0.0180
96	5.8870	0.1753	0.3906	0.1705	0.0095
100	5.8774	0.1383	0.3974	0.1307	0.0107
104	5.8155	0.1169	0.3764	0.0954	0.0070
108	5.7804	0.1132	0.3832	0.1046	0.0099
112	5.7139	0.0797	0.3761	0.0744	0.0051
116	5.6317	0.1227	0.3763	0.1287	0.0157
120	5.4691	0.1057	0.3427	0.1098	0.0115
124	5.3524	0.0414	0.4009	0.0433	0.0017
128	5.1120	0.1236	0.3625	0.1296	0.0160
132	4.9026	0.0864	0.3379	0.0876	0.0073
136	4.6705	0.0864	0.3553	0.0897	0.0075
140	4.3925	0.1019	0.3091	0.1015	0.0099
144	4.1990	0.0586	0.1812	0.0600	0.0032
148	3.8077	0.0943	0.2630	0.0950	0.0087
152	3.4612	0.1000	0.2463	0.1038	0.0102
156	3.1358	0.0418	0.1892	0.0390	0.0013
160	2.7790	0.0644	0.1880	0.0671	0.0043
164	2.3738	0.0604	0.1529	0.0585	0.0033
168	1.9695	0.0552	0.1283	0.0556	0.0030
172	1.5864	0.0425	0.0955	0.0448	0.0019
176	1.1946	0.0374	0.0760	0.0379	0.0013
180	0.7614	0.0348	0.0494	0.0312	0.0010

SPRE s/d = 16

r/R = 0.000

θ	$\tilde{u}$	u'	$\tilde{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.2611	0.0887	0.0277	0.0521	0.0025
8	0.5208	0.1081	0.0416	0.0664	0.0025
12	0.8470	0.1097	0.0706	0.0831	0.0038
16	1.2799	0.1014	0.1028	0.0865	0.0036
20	1.7498	0.1158	0.1409	0.0916	0.0052
24	2.1023	0.1003	0.1562	0.0757	0.0022
28	2.4488	0.1018	0.1808	0.0866	0.0050
32	2.8850	0.1157	0.1895	0.1121	0.0079

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
36	3.3075	0.1101	0.2178	0.1178	0.0089
40	3.6200	0.1003	0.2449	0.0932	0.0058
44	3.9137	0.1142	0.2504	0.1099	0.0072
48	4.2762	0.1248	0.2650	0.1250	0.0148
52	4.6005	0.1148	0.3105	0.1194	0.0114
54	4.7152	0.1180	0.3121	0.1096	0.0086
56	4.7907	0.1208	0.3154	0.1150	0.0055
58	4.9060	0.1245	0.3168	0.1433	0.0050
60	5.0734	0.1795	0.3312	0.1963	0.0170
62	5.1813	0.2284	0.3741	0.2287	0.0303
64	5.2912	0.1741	0.3539	0.2013	0.0073
66	5.3894	0.1662	0.3547	0.2152	0.0046
68	5.4489	0.2016	0.3923	0.2332	0.0091
70	5.4832	0.2074	0.3688	0.2707	0.0107
72	5.4942	0.2113	0.3759	0.2307	0.0078
76	5.5555	0.2243	0.3775	0.2408	0.0061
80	5.6830	0.2126	0.3980	0.2298	0.0088
84	5.7422	0.2166	0.4146	0.2160	-0.0038
88	5.7627	0.2334	0.4318	0.2205	0.0046
92	5.8317	0.1849	0.3903	0.1950	0.0130
96	5.8729	0.1585	0.3875	0.1445	0.0044
100	5.8610	0.1262	0.4005	0.1143	0.0063
104	5.8194	0.1509	0.3741	0.1520	0.0183
108	5.7719	0.0750	0.3763	0.0626	0.0028
112	5.6960	0.0825	0.3681	0.0747	0.0048
116	5.5912	0.1041	0.3593	0.1107	0.0112
120	5.4485	0.1081	0.3467	0.1159	0.0122
124	5.3169	0.0861	0.3918	0.0897	0.0076
128	5.0862	0.1097	0.3605	0.1127	0.0122
132	4.8643	0.1038	0.3094	0.1060	0.0108
136	4.6429	0.1066	0.3292	0.1080	0.0113
140	4.3832	0.1274	0.2879	0.1313	0.0165
144	4.1720	0.0822	0.1759	0.0853	0.0069
148	3.7840	0.0570	0.2451	0.0571	0.0032
152	3.4413	0.0575	0.2254	0.0573	0.0031
156	3.1240	0.0719	0.1812	0.0730	0.0052
160	2.7656	0.0194	0.1759	0.0242	0.0003
164	2.3757	0.0695	0.1344	0.0704	0.0048
168	1.9757	0.0307	0.1066	0.0311	0.0008
172	1.5881	0.0396	0.0829	0.0430	0.0015
176	1.1862	0.0477	0.0539	0.0318	0.0012
180	0.7476	0.0710	0.0216	0.0455	0.0022

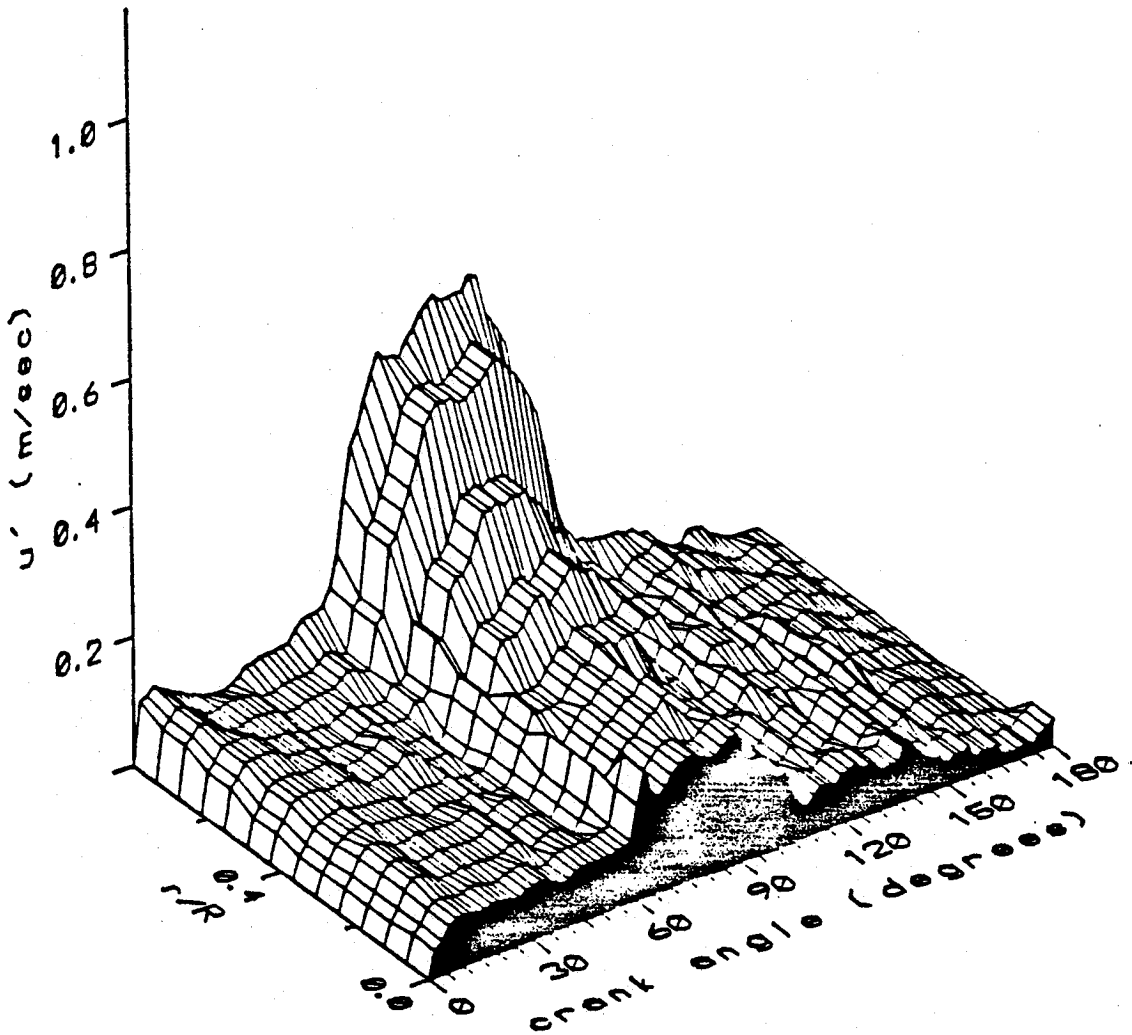


Figure 17: Streamwise velocity fluctuation at s/d = 16 (smoothed)

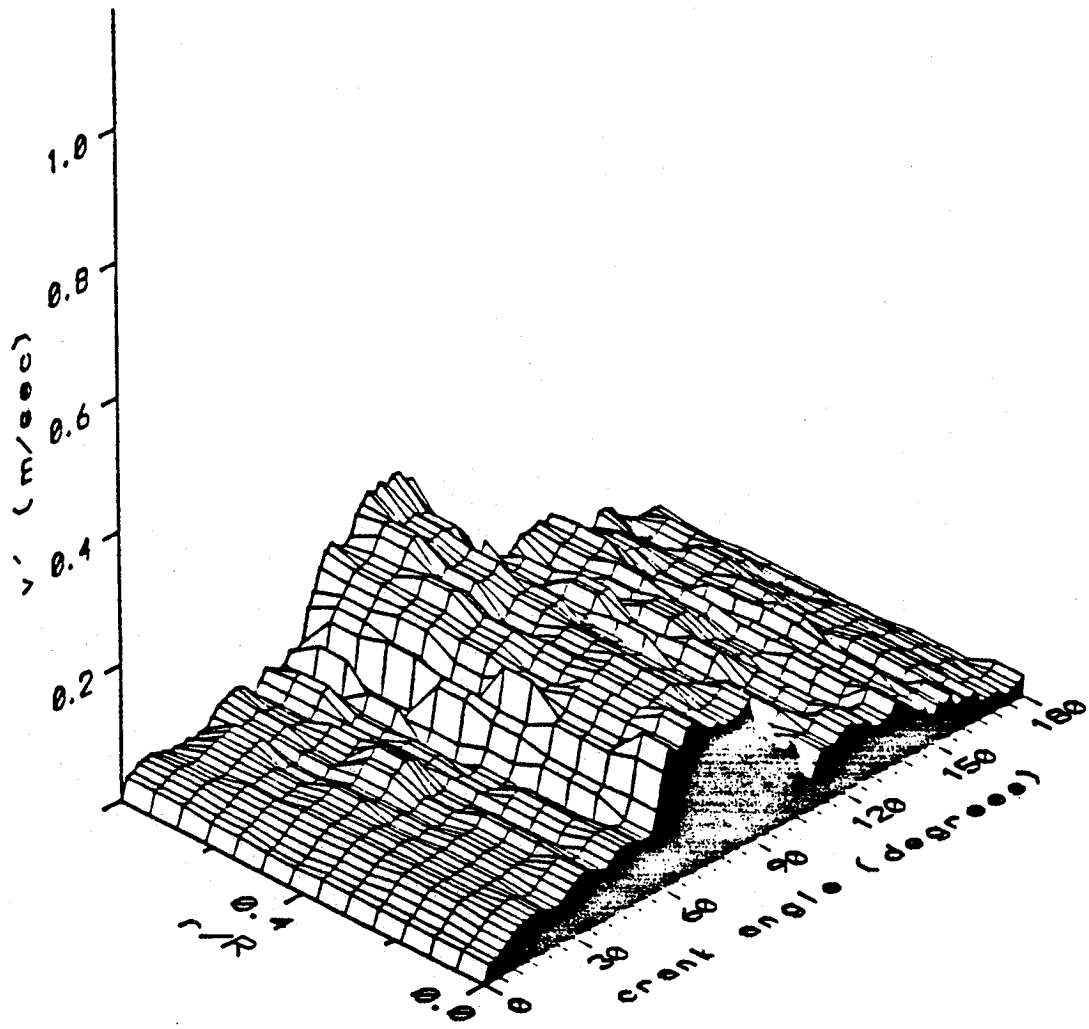


Figure 18: Radial velocity fluctuation at s/d = 16 (smoothed)

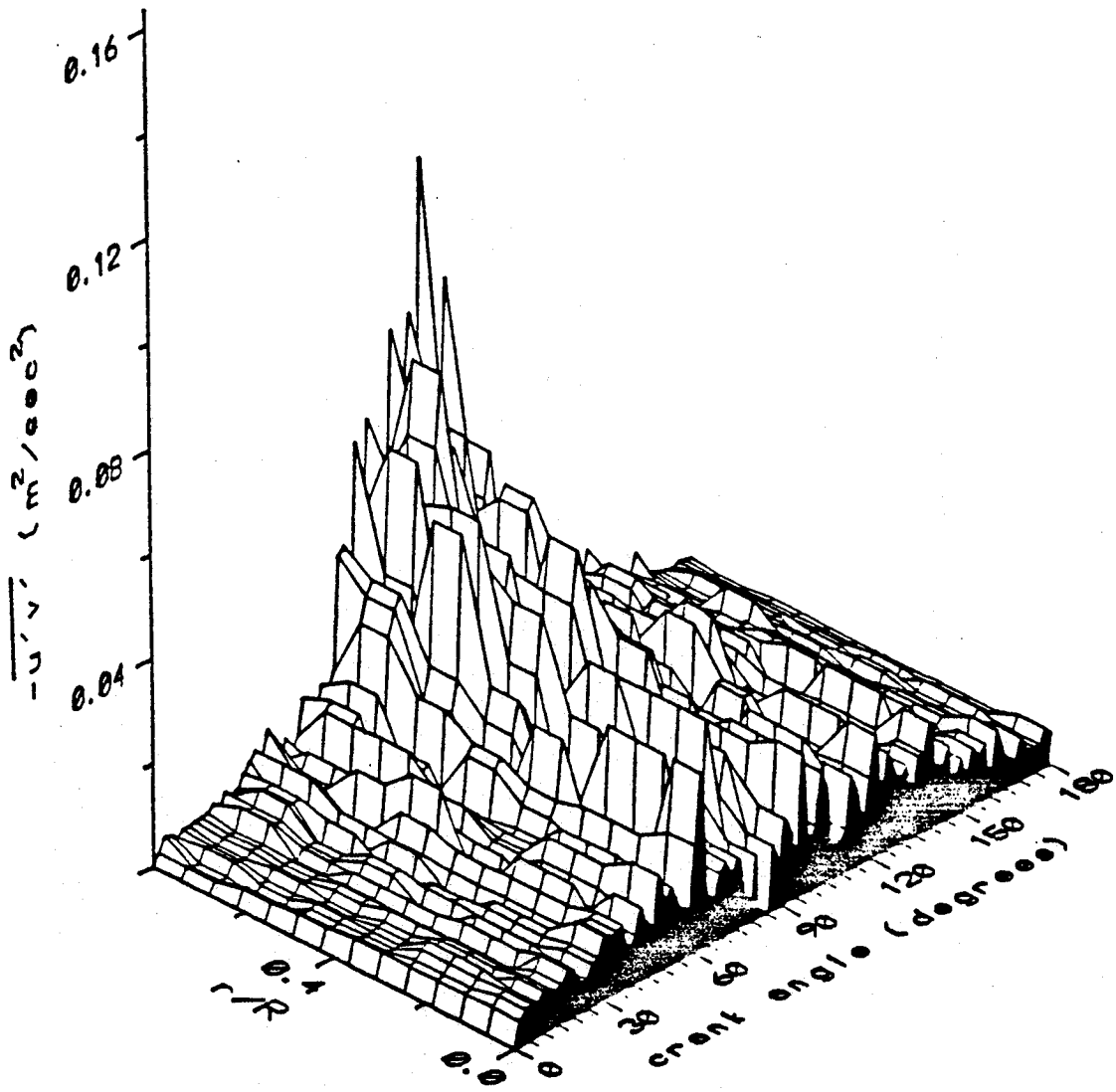


Figure 19: Reynolds shear stress at s/d = 16
(not smoothed)

CROSS-WIRE DATA

SPRE s/d = 30

r/R = 0.800

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.4472	0.1582	-0.0025	0.0535	-0.0011
8	0.7544	0.1270	0.0274	0.0669	-0.0005
12	1.0445	0.1160	0.0577	0.0645	0.0000
16	1.4388	0.1171	0.0931	0.0586	-0.0001
20	1.8399	0.0938	0.1188	0.0596	0.0009
24	2.1327	0.1146	0.1385	0.0616	0.0029
28	2.4290	0.0903	0.1701	0.0688	0.0027
32	2.8059	0.0981	0.1960	0.0640	0.0033
36	3.1315	0.1085	0.2165	0.0666	0.0046
40	3.3870	0.1258	0.2459	0.0910	0.0079
44	3.6244	0.1351	0.2737	0.0955	0.0083
48	3.9006	0.1346	0.2773	0.0763	0.0045
52	4.1960	0.1853	0.2861	0.1106	0.0108
56	4.3019	0.1799	0.3192	0.1227	0.0139
60	4.4351	0.2046	0.3273	0.1448	0.0216
64	4.6238	0.2201	0.3492	0.1487	0.0210
68	4.7714	0.1791	0.3776	0.0681	0.0034
72	4.8289	0.2047	0.3839	0.0906	0.0103
76	4.8715	0.2504	0.3936	0.1358	0.0228
78	4.9140	0.2240	0.3836	0.1153	0.0131
80	5.0142	0.3122	0.3897	0.1009	0.0107
82	5.0818	0.4065	0.3973	0.1581	0.0263
84	5.1231	0.4484	0.4279	0.1875	0.0367
86	4.9455	0.5751	0.3840	0.3075	0.0641
88	4.7351	0.6579	0.3104	0.4027	0.1464
90	4.6422	0.6116	0.2807	0.4068	0.1354
92	4.6668	0.5325	0.2957	0.3523	0.0930
94	4.7180	0.4917	0.3233	0.3044	0.0707
96	4.7995	0.4836	0.3631	0.3202	0.0937
100	4.7535	0.4552	0.3163	0.2706	0.0570
104	4.6991	0.4284	0.3210	0.2749	0.0595
108	4.5892	0.5144	0.3401	0.3204	0.0905
112	4.5499	0.4457	0.3238	0.2846	0.0610
116	4.4230	0.5059	0.3149	0.3024	0.0775
120	4.2926	0.5235	0.2835	0.3092	0.0912
124	4.1227	0.4207	0.2570	0.2898	0.0598

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
128	3.9200	0.4239	0.2444	0.2749	0.0460
132	3.6889	0.3998	0.2488	0.2404	0.0385
136	3.5521	0.4076	0.2630	0.2528	0.0394
140	3.3526	0.4202	0.2975	0.2298	0.0467
144	3.0497	0.4540	0.2254	0.2527	0.0624
148	2.7508	0.4011	0.2815	0.1981	0.0334
152	2.4668	0.4130	0.2304	0.2402	0.0458
156	2.1376	0.3857	0.1768	0.2169	0.0250
160	1.8145	0.4095	0.1697	0.2122	0.0348
164	1.5397	0.4012	0.1809	0.2064	0.0394
168	1.2162	0.3091	0.1131	0.1846	0.0208
172	0.9114	0.2678	0.0838	0.1390	0.0164
176	0.5792	0.2265	0.0709	0.1168	0.0139
180	0.3661	0.1253	0.0339	0.0697	0.0025

SPRE s/d = 30

r/R = 0.733

θ	$\tilde{u}$	u'	$\tilde{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.4109	0.1408	0.0047	0.0561	-0.0021
8	0.7039	0.1299	0.0242	0.0648	-0.0025
12	1.0161	0.1249	0.0620	0.0759	0.0005
16	1.4378	0.1165	0.0909	0.0737	0.0017
20	1.8398	0.1045	0.1286	0.0655	0.0007
24	2.1562	0.1076	0.1421	0.0736	0.0035
28	2.4580	0.0902	0.1694	0.0698	0.0028
32	2.8609	0.1289	0.2092	0.1047	0.0097
36	3.2070	0.1290	0.2233	0.1074	0.0107
40	3.4558	0.0905	0.2455	0.0680	0.0032
44	3.7183	0.0871	0.2666	0.0430	0.0018
48	4.0906	0.1407	0.2639	0.1142	0.0053
52	4.3386	0.1126	0.3064	0.1071	0.0109
56	4.4911	0.1213	0.3360	0.0582	0.0044
60	4.6867	0.1480	0.3445	0.1057	0.0101
64	4.9277	0.1506	0.3727	0.0821	0.0079
68	5.1311	0.1954	0.4234	0.1234	0.0143
72	5.2337	0.1359	0.4305	0.0576	0.0018
76	5.3096	0.1571	0.4167	0.1068	0.0079
78	5.3586	0.2285	0.4073	0.1420	0.0153
80	5.4683	0.2669	0.4014	0.1232	0.0159
82	5.4239	0.3299	0.4037	0.1166	0.0103
84	5.3493	0.4357	0.3664	0.2189	0.0507
86	5.1744	0.5865	0.3393	0.3148	0.0777
88	5.0382	0.6124	0.4138	0.3660	0.1106
90	4.9143	0.5336	0.3138	0.3727	0.0954
92	4.9120	0.5395	0.2899	0.3211	0.0888
94	5.0180	0.4704	0.3674	0.3098	0.0746
96	5.0239	0.4544	0.3291	0.2903	0.0650
100	5.0056	0.4808	0.3487	0.2767	0.0704
104	4.9230	0.4966	0.3165	0.3008	0.0673
108	4.8070	0.4502	0.3430	0.2756	0.0617
112	4.7497	0.4147	0.3147	0.2862	0.0471
116	4.6833	0.4316	0.3279	0.2749	0.0524
120	4.5307	0.4391	0.3197	0.2884	0.0735
124	4.3468	0.4324	0.2880	0.2624	0.0636
128	4.1470	0.3838	0.2858	0.2563	0.0421
132	3.9336	0.4292	0.2963	0.2224	0.0468
136	3.7714	0.3997	0.2510	0.2536	0.0434
140	3.5997	0.4218	0.2911	0.2294	0.0414
144	3.2831	0.4232	0.2461	0.2546	0.0617

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
148	3.0355	0.4353	0.2657	0.2334	0.0418
152	2.7352	0.4206	0.2647	0.2677	0.0525
156	2.4566	0.3939	0.2337	0.2562	0.0401
160	2.1026	0.4061	0.2163	0.2311	0.0333
164	1.6702	0.3546	0.1306	0.2282	0.0349
168	1.2737	0.3310	0.0979	0.1960	0.0307
172	0.9657	0.2723	0.1003	0.1736	0.0241
176	0.6303	0.2107	0.0566	0.1091	0.0086
180	0.3866	0.1415	0.0338	0.0731	0.0043

SPRE s/d = 30

r/R = 0.600

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.3650	0.1316	-0.0073	0.0557	-0.0029
8	0.6694	0.1213	0.0179	0.0685	-0.0003
12	0.9861	0.1203	0.0539	0.0723	-0.0015
16	1.4051	0.1216	0.0875	0.0738	-0.0015
20	1.8319	0.1068	0.1152	0.0779	-0.0005
24	2.1395	0.1162	0.1418	0.0779	0.0012
28	2.4491	0.1038	0.1685	0.0661	0.0014
32	2.8609	0.1028	0.2015	0.0721	0.0021
36	3.2251	0.0933	0.2211	0.0887	0.0034
40	3.5078	0.1193	0.2559	0.0855	0.0063
44	3.7683	0.0968	0.2589	0.0896	0.0063
48	4.1780	0.1364	0.2437	0.1093	0.0064
52	4.4356	0.1049	0.3255	0.0880	0.0075
56	4.6548	0.1070	0.3436	0.0909	0.0081
60	4.8532	0.0960	0.3370	0.0747	0.0051
64	5.1479	0.1193	0.3777	0.1099	0.0110
68	5.3912	0.1070	0.4063	0.1065	0.0101
72	5.4700	0.0697	0.3679	0.0464	0.0006
76	5.6279	0.1050	0.3775	0.0765	0.0054
78	5.6728	0.1595	0.3805	0.1168	0.0127
80	5.7562	0.1910	0.4000	0.1192	0.0123
82	5.7906	0.2297	0.3895	0.1344	0.0146
84	5.7150	0.3691	0.3873	0.2368	0.0406
86	5.6220	0.4726	0.3722	0.2503	0.0486

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
88	5.4040	0.5532	0.3188	0.3458	0.0930
90	5.3587	0.4818	0.3353	0.3343	0.0729
92	5.3024	0.4633	0.3308	0.2844	0.0634
94	5.3414	0.4463	0.3621	0.2837	0.0624
96	5.3836	0.4195	0.3845	0.2549	0.0536
100	5.3918	0.4136	0.3696	0.2237	0.0382
104	5.3321	0.4444	0.3858	0.2900	0.0670
108	5.2162	0.4244	0.3928	0.2644	0.0523
112	5.1289	0.4493	0.3680	0.2748	0.0569
116	4.9577	0.4806	0.3507	0.2899	0.0758
120	4.8443	0.4096	0.3427	0.2785	0.0733
124	4.6796	0.3906	0.3377	0.2583	0.0546
128	4.4502	0.3622	0.3001	0.2330	0.0376
132	4.3230	0.3554	0.2856	0.2131	0.0372
136	4.0886	0.3996	0.2809	0.2277	0.0487
140	3.9401	0.3682	0.2885	0.2303	0.0316
144	3.7346	0.3717	0.2574	0.1675	0.0259
148	3.4596	0.3160	0.2778	0.2022	0.0235
152	3.0764	0.4170	0.2518	0.2349	0.0345
156	2.7873	0.3501	0.2119	0.1939	0.0320
160	2.3643	0.4025	0.2126	0.2083	0.0302
164	1.8936	0.3794	0.1527	0.2207	0.0379
168	1.5529	0.3106	0.1557	0.1878	0.0283
172	1.1514	0.2891	0.0929	0.1830	0.0183
176	0.7916	0.2502	0.0760	0.1377	0.0132
180	0.4450	0.1506	0.0450	0.0755	0.0041

SPRE s/d = 30

r/R = 0.467

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.3205	0.1028	0.0114	0.0528	0.0001
8	0.6029	0.1402	0.0245	0.0750	-0.0017
12	0.9175	0.1278	0.0618	0.0791	-0.0003
16	1.3359	0.1191	0.0797	0.0806	-0.0024
20	1.7636	0.1066	0.1237	0.0790	-0.0006
24	2.0766	0.1117	0.1384	0.0828	0.0012

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
28	2.4099	0.1283	0.1762	0.1042	0.0064
32	2.8077	0.1130	0.2062	0.0737	0.0004
36	3.1927	0.0989	0.2148	0.0821	0.0029
40	3.4773	0.0751	0.2397	0.0621	0.0005
44	3.7743	0.1043	0.2634	0.0713	0.0036
48	4.1665	0.1303	0.2346	0.1212	0.0103
52	4.4197	0.0966	0.3165	0.0852	0.0058
56	4.6201	0.1139	0.3175	0.1053	0.0092
60	4.8697	0.1077	0.3365	0.0970	0.0091
64	5.2193	0.1193	0.4151	0.1015	0.0096
68	5.3653	0.0639	0.3660	0.0555	0.0017
72	5.4971	0.1156	0.3638	0.1192	0.0124
76	5.6428	0.1086	0.3760	0.0946	0.0080
78	5.7177	0.1610	0.3866	0.1443	0.0176
80	5.7893	0.1341	0.3900	0.1062	0.0056
82	5.7946	0.1907	0.3814	0.1524	0.0117
84	5.8187	0.3063	0.3866	0.2234	0.0419
86	5.8584	0.3754	0.3724	0.2145	0.0357
88	5.7459	0.4571	0.3397	0.3329	0.0817
90	5.6090	0.4564	0.3303	0.2596	0.0509
92	5.5576	0.4334	0.3283	0.2617	0.0538
94	5.5854	0.4331	0.3920	0.2611	0.0524
96	5.6153	0.4196	0.4119	0.3042	0.0672
100	5.5797	0.4401	0.3515	0.2347	0.0496
104	5.5452	0.3628	0.3805	0.2372	0.0310
108	5.4114	0.4168	0.3451	0.2550	0.0443
112	5.3362	0.4060	0.4009	0.2467	0.0440
116	5.2502	0.4208	0.3734	0.2740	0.0516
120	5.0956	0.3748	0.3556	0.2410	0.0433
124	4.9263	0.3393	0.3459	0.1888	0.0222
128	4.7605	0.3185	0.3219	0.2321	0.0355
132	4.5040	0.3321	0.2790	0.2447	0.0479
136	4.3394	0.2771	0.2771	0.1941	0.0244
140	4.1706	0.2583	0.2654	0.1366	0.0152
144	3.9240	0.2671	0.2714	0.1715	0.0207
148	3.6300	0.2655	0.2810	0.1886	0.0199
152	3.2796	0.3353	0.2571	0.1635	0.0096
156	3.0164	0.2520	0.2362	0.1860	0.0133
160	2.6084	0.2833	0.2028	0.1808	0.0188
164	2.1309	0.3147	0.1706	0.2369	0.0226
168	1.7169	0.2874	0.1544	0.2041	0.0217
172	1.2689	0.2771	0.0909	0.1870	0.0151
176	0.8284	0.2428	0.0773	0.1459	0.0160
180	0.4829	0.1578	0.0467	0.0908	0.0058

SPRE s/d = 30

r/R = 0.333

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.2747	0.0773	0.0170	0.0517	0.0004
8	0.5389	0.1138	0.0285	0.0682	0.0002
12	0.8817	0.1210	0.0633	0.0840	-0.0012
16	1.3005	0.0945	0.0960	0.0818	-0.0002
20	1.7403	0.1124	0.1198	0.0792	-0.0020
24	2.0697	0.1188	0.1536	0.0833	0.0012
28	2.3738	0.1106	0.1681	0.0928	0.0036
32	2.7912	0.1058	0.1855	0.0879	0.0031
36	3.1939	0.0943	0.2100	0.0706	0.0009
40	3.4889	0.0876	0.2487	0.0714	0.0023
44	3.7725	0.0831	0.2567	0.0649	0.0015
48	4.1719	0.1179	0.2444	0.1276	0.0058
52	4.4498	0.1189	0.3099	0.1253	0.0125
56	4.6827	0.1145	0.3444	0.1134	0.0108
60	4.9046	0.1151	0.3318	0.1143	0.0116
64	5.2017	0.1096	0.3805	0.1047	0.0097
68	5.4455	0.1390	0.4014	0.1439	0.0187
72	5.5465	0.0993	0.3555	0.1028	0.0090
76	5.6672	0.1239	0.3551	0.1247	0.0134
78	5.7597	0.1519	0.3862	0.1455	0.0178
80	5.8211	0.1324	0.3661	0.1035	0.0057
82	5.9025	0.1654	0.3943	0.1561	0.0123
84	5.9333	0.2447	0.3880	0.1710	0.0212
86	6.0092	0.3272	0.4001	0.2218	0.0287
88	6.0573	0.3902	0.3616	0.2818	0.0558
90	5.9271	0.4199	0.3685	0.2695	0.0432
92	5.8480	0.4349	0.3715	0.2927	0.0589
94	5.8083	0.3824	0.3515	0.2687	0.0415
96	5.8178	0.3973	0.3839	0.2616	0.0380
100	5.8178	0.3463	0.3870	0.2521	0.0393
104	5.7328	0.3474	0.3677	0.2247	0.0276
108	5.5789	0.3620	0.3716	0.2202	0.0248
112	5.5170	0.3550	0.3846	0.2094	0.0173
116	5.3807	0.3737	0.3638	0.2389	0.0318
120	5.2974	0.3395	0.3757	0.2324	0.0285
124	5.1116	0.3253	0.3737	0.2380	0.0274
128	4.9624	0.3361	0.3321	0.2502	0.0529
132	4.7395	0.2370	0.3175	0.1765	0.0163
136	4.5120	0.1551	0.3077	0.1328	0.0031
140	4.2960	0.1733	0.2466	0.1559	0.0123
144	4.0142	0.1714	0.2756	0.1376	0.0050
148	3.7400	0.2042	0.2498	0.1412	0.0153

θ	$\tilde{u}$	u'	$\tilde{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
152	3.4052	0.2553	0.2371	0.1921	0.0020
156	3.0659	0.2561	0.1755	0.1859	0.0163
160	2.6861	0.2671	0.1967	0.1783	0.0083
164	2.2303	0.2563	0.1636	0.1890	0.0098
168	1.8339	0.2664	0.1318	0.1964	0.0133
172	1.3825	0.2690	0.1189	0.1872	0.0141
176	0.9456	0.2342	0.0931	0.1566	0.0156
180	0.5111	0.1592	0.0438	0.0904	0.0064

SPRE s/d = 30

r/R = 0.200

θ	$\tilde{u}$	u'	$\tilde{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.2644	0.0753	0.0218	0.0501	0.0014
8	0.5110	0.1216	0.0313	0.0718	0.0021
12	0.8520	0.1118	0.0763	0.0810	0.0009
16	1.2527	0.0999	0.1060	0.0696	0.0000
20	1.6710	0.1106	0.1252	0.0908	0.0042
24	2.0262	0.1026	0.1525	0.0835	0.0021
28	2.3639	0.1014	0.1676	0.0984	0.0037
32	2.7835	0.1051	0.2009	0.0932	0.0038
36	3.1659	0.1183	0.2291	0.1118	0.0085
40	3.4537	0.0819	0.2481	0.0693	0.0013
44	3.7378	0.1271	0.2473	0.1182	0.0119
48	4.1524	0.1244	0.2592	0.1066	0.0040
52	4.4225	0.0867	0.3147	0.0892	0.0056
56	4.6495	0.0717	0.3371	0.0652	0.0026
60	4.8774	0.0702	0.3293	0.0583	0.0027
64	5.1664	0.0890	0.3643	0.0805	0.0054
68	5.4158	0.1679	0.3952	0.1714	0.0274
72	5.5226	0.0764	0.3542	0.0715	0.0043
76	5.6409	0.1197	0.3632	0.1274	0.0128
78	5.7144	0.0962	0.3622	0.1045	0.0048
80	5.7977	0.1372	0.3539	0.1307	0.0072
82	5.8813	0.2007	0.3614	0.1842	0.0219
84	5.9653	0.1950	0.3516	0.1764	0.0147
86	6.1590	0.3250	0.3889	0.2476	0.0399
88	6.2116	0.3061	0.3591	0.2553	0.0237

θ	$\tilde{u}$	u'	$\tilde{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
90	6.0627	0.4075	0.3777	0.2748	0.0455
92	5.9538	0.3101	0.3984	0.2157	0.0236
94	5.8587	0.3355	0.3580	0.2512	0.0267
96	5.9027	0.2877	0.3970	0.2233	0.0239
100	5.8501	0.3174	0.3812	0.2174	0.0209
104	5.7968	0.3006	0.3835	0.2048	0.0144
108	5.6861	0.2882	0.3488	0.1988	0.0154
112	5.6043	0.3274	0.3608	0.2593	0.0259
116	5.4799	0.2893	0.3526	0.2218	0.0118
120	5.3979	0.2773	0.3274	0.1911	0.0109
124	5.2439	0.2864	0.3370	0.2234	0.0288
128	5.0158	0.2725	0.3193	0.2182	0.0251
132	4.7339	0.2206	0.2782	0.1725	0.0168
136	4.4914	0.1946	0.2864	0.1538	0.0151
140	4.2754	0.1929	0.2455	0.1491	0.0152
144	3.9894	0.2034	0.2318	0.1471	0.0077
148	3.7441	0.2105	0.2297	0.1507	0.0167
152	3.4030	0.2226	0.2122	0.1478	0.0019
156	3.0378	0.2683	0.1620	0.1734	0.0125
160	2.6719	0.2713	0.1521	0.1942	0.0096
164	2.2520	0.2536	0.1275	0.1957	0.0120
168	1.8411	0.2625	0.0801	0.2016	0.0133
172	1.4337	0.2665	0.1012	0.1655	0.0120
176	0.9965	0.2252	0.0821	0.1574	0.0122
180	0.5649	0.1622	0.0312	0.0937	0.0034

SPRE s/d = 30

r/R = 0.000

θ	$\tilde{u}$	u'	$\tilde{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.2584	0.0752	0.0222	0.0513	0.0018
8	0.5016	0.1182	0.0451	0.0760	0.0042
12	0.8136	0.1166	0.0764	0.0872	0.0043
16	1.2239	0.1025	0.0952	0.0878	0.0030
20	1.6723	0.0954	0.1348	0.0890	0.0024
24	1.9864	0.1060	0.1445	0.0866	0.0036
28	2.3326	0.1074	0.1647	0.1007	0.0055
32	2.7505	0.0916	0.1979	0.0886	0.0042
36	3.1388	0.0914	0.2129	0.0812	0.0040

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
40	3.4487	0.1080	0.2485	0.1007	0.0071
44	3.7207	0.0863	0.2510	0.0834	0.0047
48	4.1216	0.1263	0.2347	0.1297	0.0071
52	4.3848	0.0905	0.3076	0.0946	0.0067
56	4.6113	0.0744	0.3164	0.0730	0.0034
60	4.8446	0.1154	0.3150	0.1187	0.0118
64	5.1395	0.1000	0.3536	0.0986	0.0079
68	5.3685	0.0985	0.3668	0.1046	0.0086
72	5.4744	0.0824	0.3235	0.0780	0.0052
76	5.6074	0.0835	0.3393	0.0778	0.0033
78	5.6860	0.1027	0.3495	0.1000	0.0049
80	5.7940	0.1374	0.3668	0.1593	0.0121
82	5.8741	0.1792	0.3751	0.1688	0.0124
84	5.9850	0.2422	0.3616	0.1728	0.0133
86	6.1766	0.2992	0.3565	0.2316	0.0191
88	6.2618	0.3130	0.3575	0.2350	0.0140
90	6.1204	0.3879	0.3537	0.2201	0.0203
92	5.9615	0.3266	0.3672	0.2422	0.0113
94	5.8811	0.3143	0.3546	0.2508	0.0193
96	5.8453	0.2822	0.3577	0.2005	-0.0131
100	5.8362	0.2909	0.3682	0.2112	0.0063
104	5.7415	0.2763	0.3646	0.2096	-0.0020
108	5.6736	0.2805	0.3434	0.2105	0.0114
112	5.5963	0.2922	0.3443	0.2149	0.0019
116	5.5247	0.3121	0.3363	0.2130	0.0063
120	5.3931	0.2772	0.3658	0.2241	0.0076
124	5.2336	0.2391	0.3494	0.1755	0.0061
128	4.9574	0.2266	0.2810	0.1574	0.0076
132	4.6761	0.2311	0.2387	0.1748	0.0152
136	4.3907	0.2411	0.2384	0.1527	0.0142
140	4.1317	0.2048	0.1760	0.1269	-0.0004
144	3.8399	0.2667	0.1968	0.1619	-0.0044
148	3.5455	0.2333	0.1876	0.1465	0.0056
152	3.1437	0.2955	0.1624	0.2093	0.0054
156	2.8431	0.2996	0.1350	0.1843	0.0026
160	2.5168	0.3121	0.1037	0.2007	-0.0016
164	2.1393	0.2584	0.0892	0.1633	-0.0019
168	1.7057	0.2821	0.0718	0.1772	0.0026
172	1.4133	0.2532	0.0859	0.1836	-0.0006
176	0.9781	0.2090	0.0792	0.1334	0.0006
180	0.5599	0.1656	0.0313	0.0918	-0.0002

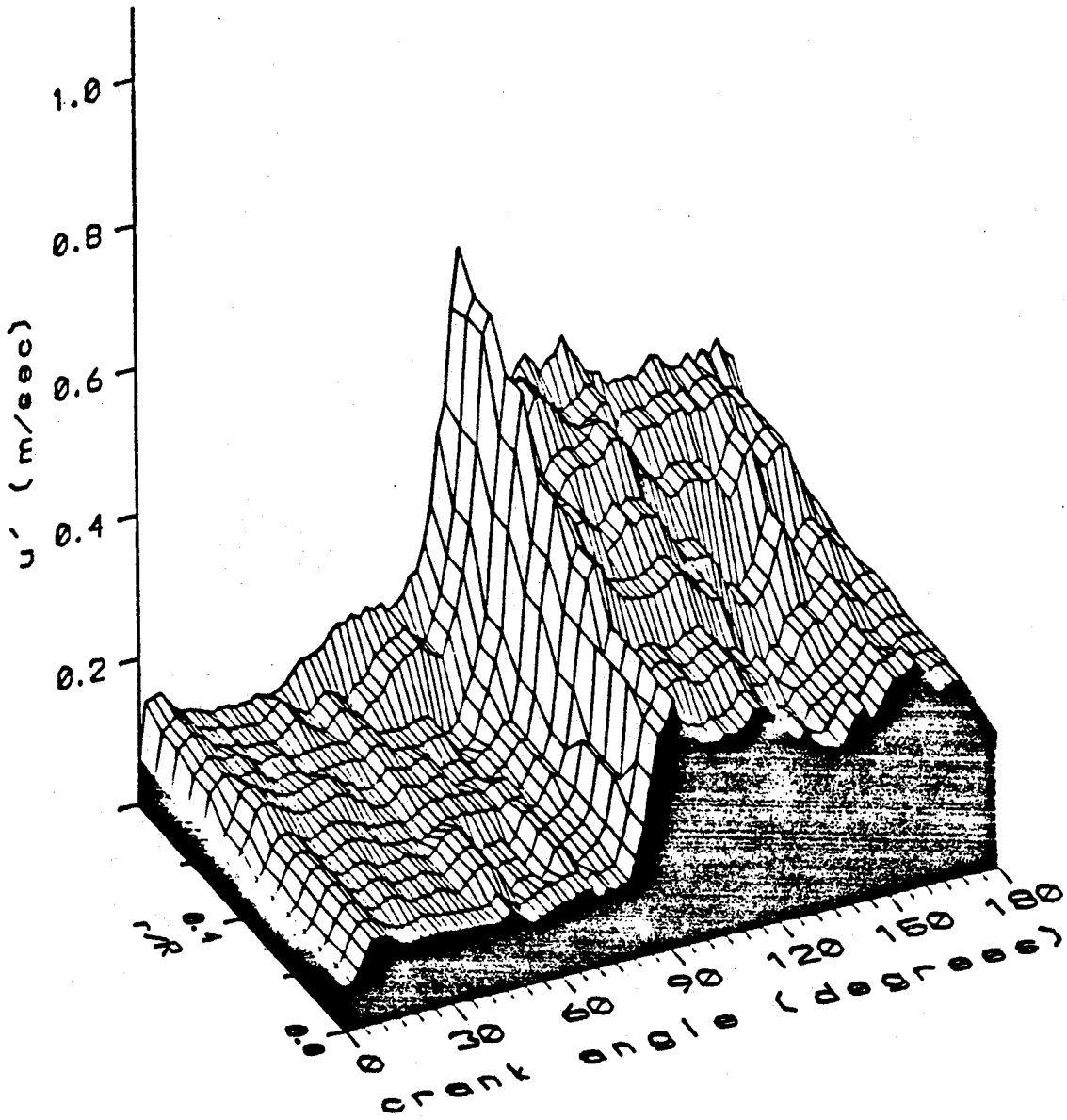


Figure 20: Streamwise velocity fluctuation at s/d = 30 (smoothed)

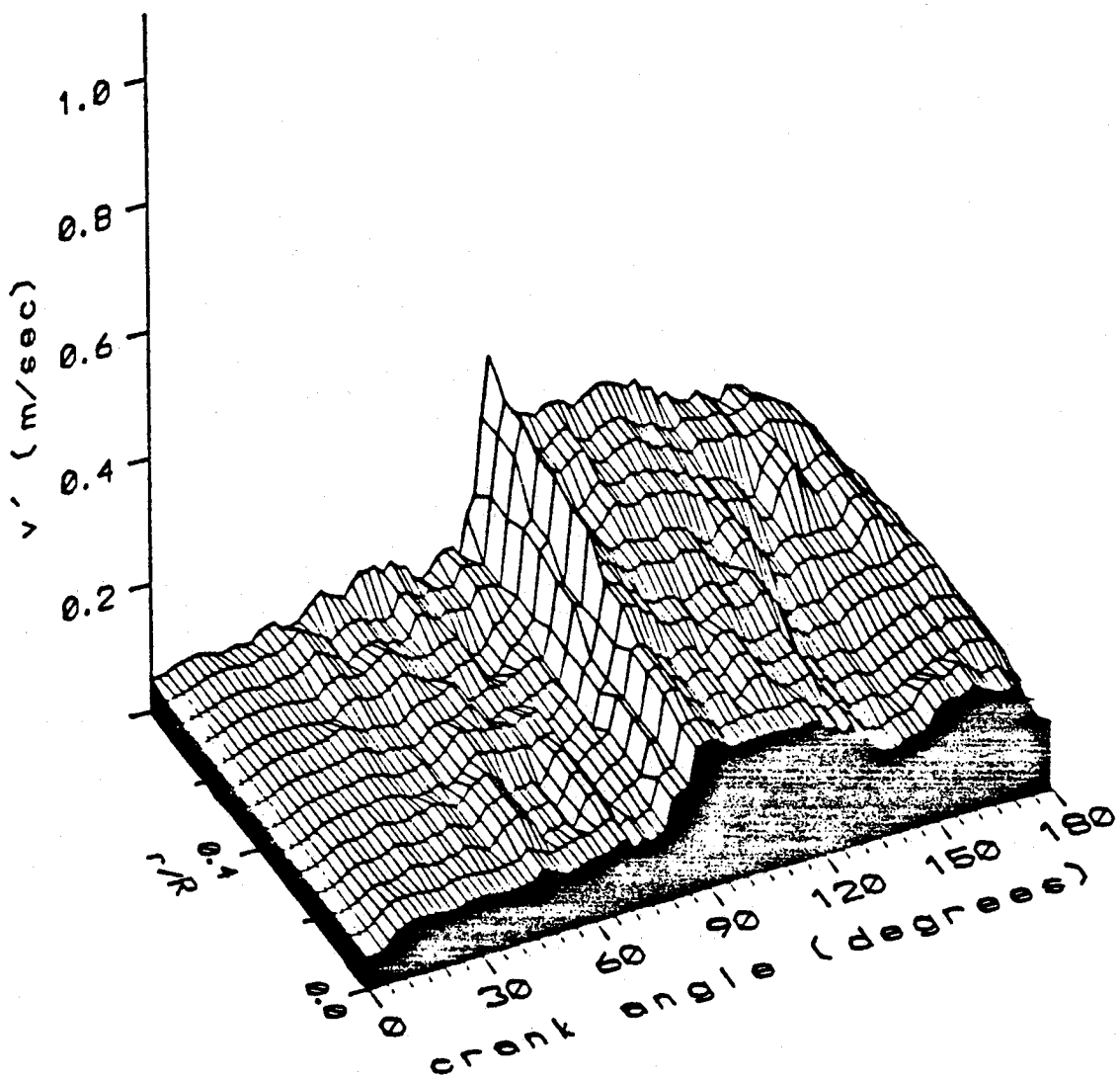


Figure 21: Radial velocity fluctuation at s/d = 30 (smoothed)

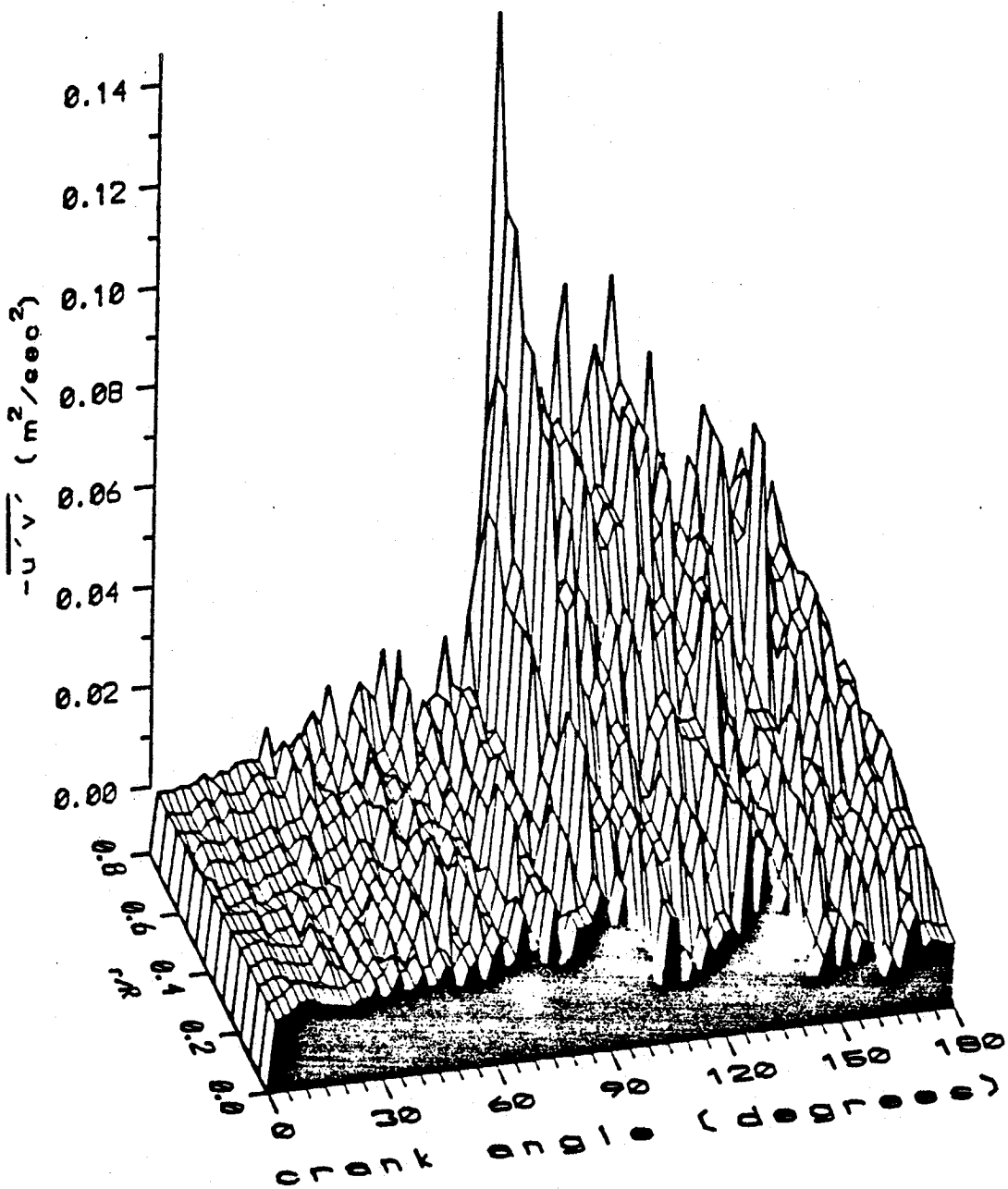


Figure 22: Reynolds shear stress at s/d = 30 (not smoothed)

CROSS-WIRE DATA

SPRE s/d = 44

r/R = 0.800

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.4522	0.1425	-0.0103	0.0561	-0.0009
8	0.7957	0.1309	0.0183	0.0651	-0.0009
12	1.1081	0.1198	0.0568	0.0680	-0.0003
16	1.4820	0.1282	0.0882	0.0767	0.0033
20	1.8827	0.1033	0.1051	0.0649	0.0018
24	2.2207	0.1111	0.1416	0.0670	0.0035
28	2.5357	0.1106	0.1664	0.0657	0.0039
32	2.8830	0.1169	0.1849	0.0844	0.0063
36	3.2464	0.1201	0.2193	0.0788	0.0067
40	3.5073	0.1037	0.2400	0.0424	0.0015
44	3.7303	0.1397	0.2462	0.1031	0.0093
48	4.0460	0.1724	0.2845	0.1063	0.0120
52	4.3070	0.1700	0.2719	0.1032	0.0085
56	4.4893	0.1749	0.3250	0.1399	0.0195
60	4.6331	0.2121	0.3291	0.1610	0.0268
64	4.8253	0.2185	0.3565	0.1403	0.0199
68	4.9883	0.1760	0.3700	0.0591	0.0034
72	5.0692	0.1872	0.3762	0.0934	0.0080
76	5.1513	0.2192	0.3943	0.1427	0.0217
80	5.2289	0.2130	0.4039	0.1211	0.0162
84	5.2998	0.2046	0.4281	0.1093	0.0150
88	5.2533	0.2104	0.4011	0.1163	0.0132
92	5.2287	0.2094	0.4191	0.1091	0.0111
96	5.1776	0.2287	0.4165	0.0921	0.0104
98	5.1492	0.3066	0.3917	0.1197	0.0157
100	5.1589	0.4313	0.3912	0.1399	0.0191
102	5.1818	0.5525	0.3824	0.1668	0.0355
104	5.0879	0.6209	0.3474	0.3430	0.0860
106	4.7442	0.7768	0.3365	0.4172	0.1789
108	4.6660	0.6386	0.2398	0.3881	0.1310
110	4.6148	0.5137	0.2615	0.3532	0.1121
112	4.6831	0.4254	0.3242	0.2943	0.0664
114	4.6592	0.4379	0.3254	0.2791	0.0768
116	4.5754	0.4624	0.2874	0.2827	0.0737
120	4.4687	0.4630	0.2729	0.2850	0.0762
124	4.2101	0.4240	0.2624	0.2669	0.0427

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
128	4.0531	0.4195	0.2258	0.2700	0.0436
132	3.8974	0.4415	0.2521	0.2726	0.0435
136	3.6958	0.4080	0.2669	0.3071	0.0567
140	3.3610	0.4096	0.2051	0.2699	0.0449
144	3.1637	0.3974	0.2426	0.2445	0.0366
148	2.8658	0.3741	0.2129	0.2125	0.0358
152	2.5217	0.3492	0.1610	0.2194	0.0413
156	2.1675	0.3270	0.1354	0.1948	0.0295
160	1.9183	0.3058	0.1341	0.1750	0.0232
164	1.5690	0.3181	0.1265	0.1602	0.0282
168	1.2111	0.2461	0.0918	0.1262	0.0133
172	0.8839	0.2358	0.0754	0.1085	0.0119
176	0.5250	0.1957	0.0462	0.0795	0.0068
180	0.3196	0.1044	0.0283	0.0470	0.0025

SPRE s/d = 44

r/R = 0.733

θ	$\tilde{u}$	u'	$\tilde{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.3766	0.1397	-0.0151	0.0553	-0.0021
8	0.7355	0.1431	0.0141	0.0584	-0.0014
12	1.0757	0.1198	0.0465	0.0726	-0.0017
16	1.4597	0.1127	0.0767	0.0755	0.0008
20	1.8821	0.1109	0.1042	0.0629	0.0008
24	2.2217	0.1189	0.1329	0.0827	0.0040
28	2.5460	0.1085	0.1539	0.0742	0.0041
32	2.9246	0.0878	0.1835	0.0526	0.0006
36	3.3112	0.1389	0.2108	0.1248	0.0134
40	3.6087	0.1087	0.2412	0.0747	0.0045
44	3.8658	0.1003	0.2642	0.0679	0.0043
48	4.2126	0.1276	0.2320	0.0908	0.0010
52	4.4675	0.1130	0.2933	0.0925	0.0089
56	4.7120	0.1567	0.3454	0.1407	0.0191
60	4.8585	0.1547	0.3190	0.1206	0.0136
64	5.1076	0.1377	0.3571	0.0915	0.0094
68	5.3232	0.1637	0.3979	0.1017	0.0107
72	5.4630	0.1336	0.4082	0.0870	0.0058
76	5.5169	0.1786	0.3834	0.1473	0.0172
80	5.5942	0.1305	0.3719	0.0796	0.0031
84	5.6733	0.1483	0.3888	0.0750	0.0037
88	5.6932	0.1843	0.3894	0.1176	0.0118
92	5.6883	0.1717	0.4000	0.0962	0.0076
96	5.6972	0.2291	0.3965	0.1491	0.0212
98	5.7133	0.3063	0.4051	0.1901	0.0306
100	5.6726	0.3401	0.3867	0.1378	0.0193
102	5.5948	0.4601	0.3803	0.2083	0.0349
104	5.4581	0.6025	0.3901	0.2974	0.1059
106	5.1104	0.6002	0.3209	0.4185	0.1155
108	4.8957	0.6338	0.2876	0.3847	0.1139
110	4.9102	0.5478	0.2430	0.3369	0.1038
112	4.8994	0.4832	0.3146	0.2620	0.0711
114	4.8174	0.4246	0.3317	0.2927	0.0663
116	4.7651	0.4045	0.3048	0.2931	0.0566
120	4.5942	0.4164	0.2925	0.2520	0.0581
124	4.4716	0.3863	0.2704	0.2635	0.0507
128	4.2346	0.4365	0.2310	0.3004	0.0717
132	4.0133	0.3669	0.2301	0.2485	0.0278
136	3.8312	0.3737	0.2419	0.2521	0.0434
140	3.5161	0.3429	0.2024	0.2321	0.0356
144	3.3057	0.3730	0.2387	0.2108	0.0428
148	3.0008	0.3249	0.1779	0.2033	0.0326
152	2.6712	0.3288	0.1782	0.2081	0.0332

θ	$\tilde{u}$	u'	$\tilde{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
156	2.3349	0.3079	0.1734	0.1962	0.0330
160	2.0123	0.2822	0.1305	0.2123	0.0315
164	1.6478	0.2675	0.1025	0.1660	0.0252
168	1.3010	0.2525	0.1131	0.1468	0.0164
172	0.9278	0.2192	0.0653	0.1027	0.0093
176	0.5821	0.1910	0.0465	0.0819	0.0092
180	0.3475	0.1065	0.0279	0.0546	0.0025

SPRE s/d = 44

r/R = 0.600

θ	$\tilde{u}$	u'	$\tilde{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.3368	0.1126	-0.0088	0.0536	-0.0018
8	0.6816	0.1334	0.0160	0.0654	-0.0016
12	1.0089	0.1206	0.0424	0.0774	-0.0011
16	1.4312	0.1155	0.0742	0.0727	-0.0013
20	1.8617	0.1215	0.1090	0.0864	-0.0007
24	2.2087	0.1158	0.1301	0.0986	0.0033
28	2.5492	0.1132	0.1513	0.0805	0.0027
32	2.9236	0.1019	0.1989	0.0620	0.0014
36	3.3001	0.0885	0.2152	0.0820	0.0037
40	3.5957	0.1172	0.2285	0.1041	0.0073
44	3.8861	0.0986	0.2487	0.0907	0.0061
48	4.2718	0.1144	0.2305	0.1041	0.0057
52	4.5285	0.0944	0.3101	0.0913	0.0060
56	4.7805	0.1315	0.3390	0.1329	0.0147
60	4.9927	0.1202	0.3470	0.1250	0.0136
64	5.2216	0.1287	0.3615	0.1092	0.0123
68	5.4791	0.0756	0.3899	0.0683	0.0036
72	5.5878	0.0909	0.3588	0.0875	0.0059
76	5.6943	0.0959	0.3517	0.0784	0.0060
80	5.8294	0.1315	0.3603	0.1161	0.0119
84	5.9651	0.1630	0.3813	0.1533	0.0218
88	6.0096	0.1081	0.3823	0.0712	0.0044
92	6.0451	0.1357	0.3838	0.1134	0.0117
96	6.0782	0.1682	0.3786	0.1238	0.0141
98	6.0531	0.2026	0.3825	0.0788	0.0051
100	6.0032	0.2818	0.3779	0.1289	0.0156
102	5.9468	0.2793	0.3776	0.1955	0.0243

θ	$\tilde{u}$	u'	$\tilde{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
104	5.7473	0.4379	0.3715	0.2661	0.0550
106	5.5184	0.5940	0.3721	0.3302	0.1067
108	5.2993	0.5320	0.2848	0.3494	0.1014
110	5.1873	0.4744	0.2970	0.2801	0.0503
112	5.1791	0.4059	0.3874	0.2928	0.0585
114	5.0902	0.3955	0.3299	0.2567	0.0566
116	5.0281	0.3967	0.3605	0.2660	0.0517
120	4.8540	0.4199	0.3362	0.2551	0.0522
124	4.7375	0.4294	0.3268	0.2882	0.0705
128	4.5353	0.3511	0.3172	0.2511	0.0392
132	4.2885	0.3635	0.2951	0.2784	0.0583
136	4.0664	0.3393	0.2521	0.2350	0.0313
140	3.7926	0.3397	0.2351	0.2323	0.0359
144	3.5485	0.3311	0.2581	0.2136	0.0341
148	3.2446	0.2997	0.2215	0.2257	0.0298
152	2.8621	0.3358	0.1908	0.2114	0.0382
156	2.5617	0.2588	0.1812	0.1726	0.0189
160	2.1643	0.2805	0.1361	0.1867	0.0276
164	1.8231	0.2545	0.1142	0.1479	0.0201
168	1.4367	0.2177	0.1085	0.1520	0.0190
172	1.0746	0.2070	0.0823	0.1184	0.0112
176	0.6787	0.1862	0.0586	0.0950	0.0097
180	0.3800	0.1135	0.0343	0.0593	0.0042

SPRE s/d = 44

r/R = 0.467

θ	$\tilde{u}$	u'	$\tilde{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.3056	0.0956	0.0038	0.0527	-0.0008
8	0.6161	0.1133	0.0154	0.0796	0.0001
12	0.9797	0.1336	0.0468	0.0751	0.0005
16	1.3836	0.1178	0.0817	0.0753	-0.0014
20	1.8341	0.1303	0.1105	0.0920	0.0030
24	2.2036	0.1277	0.1429	0.0890	0.0011
28	2.5305	0.1204	0.1625	0.0961	0.0034
32	2.9152	0.0962	0.1959	0.0743	-0.0003
36	3.3242	0.1130	0.2205	0.0986	0.0049
40	3.6276	0.1201	0.2537	0.1042	0.0066
44	3.9263	0.0934	0.2564	0.0887	0.0037
48	4.2970	0.0946	0.2312	0.0843	0.0050
52	4.5842	0.1076	0.3178	0.1035	0.0084

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
56	4.8104	0.0845	0.3101	0.0797	0.0041
60	5.0541	0.1114	0.3483	0.1121	0.0103
64	5.3401	0.1001	0.3961	0.0935	0.0073
68	5.5347	0.0869	0.3576	0.0802	0.0048
72	5.6648	0.0672	0.3466	0.0658	0.0036
76	5.8058	0.1163	0.3738	0.1145	0.0121
80	5.9636	0.1216	0.3783	0.1227	0.0128
84	6.0882	0.0927	0.3845	0.0933	0.0081
88	6.1584	0.0672	0.4009	0.0579	0.0027
92	6.2206	0.1527	0.4049	0.1440	0.0200
96	6.2290	0.1199	0.3926	0.0901	0.0056
98	6.2016	0.2243	0.3771	0.1893	0.0300
100	6.1673	0.2247	0.3658	0.1657	0.0183
102	6.0698	0.2853	0.3712	0.2071	0.0240
104	6.0420	0.3569	0.3620	0.2736	0.0434
106	6.0827	0.4551	0.4105	0.3467	0.0799
108	5.7450	0.5014	0.3478	0.3159	0.0840
110	5.5130	0.4016	0.3216	0.2880	0.0610
112	5.3613	0.3571	0.3579	0.2487	0.0278
114	5.3778	0.3114	0.3743	0.2449	0.0170
116	5.2977	0.3538	0.3618	0.2282	0.0261
120	5.1242	0.3628	0.3394	0.2713	0.0498
124	4.9753	0.3573	0.3615	0.2389	0.0418
128	4.7732	0.3359	0.3228	0.2350	0.0328
132	4.5402	0.3080	0.2923	0.2400	0.0418
136	4.3106	0.2991	0.2567	0.2080	0.0341
140	3.9680	0.3467	0.2236	0.2387	0.0379
144	3.7548	0.2999	0.2361	0.2261	0.0285
148	3.4093	0.3207	0.2024	0.2104	0.0352
152	3.0723	0.2583	0.2154	0.1667	0.0199
156	2.7081	0.2636	0.1929	0.1942	0.0216
160	2.3470	0.2626	0.1428	0.1747	0.0191
164	1.9850	0.2446	0.1511	0.1637	0.0239
168	1.5479	0.2395	0.1041	0.1365	0.0166
172	1.1941	0.2064	0.0983	0.1294	0.0138
176	0.7893	0.1743	0.0512	0.0980	0.0077
180	0.4362	0.1118	0.0371	0.0675	0.0036

SPRE s/d = 44

r/R = 0.333

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.2980	0.0961	0.0060	0.0623	-0.0002
8	0.5851	0.1277	0.0270	0.0745	0.0008
12	0.9228	0.1259	0.0450	0.0780	0.0003
16	1.3440	0.1076	0.0768	0.0867	-0.0010
20	1.7855	0.1235	0.1020	0.0776	0.0011
24	2.1405	0.1013	0.1288	0.0761	-0.0012
28	2.4729	0.1337	0.1519	0.0983	0.0040
32	2.8825	0.1116	0.1811	0.0815	0.0017
36	3.2880	0.1125	0.2149	0.0954	0.0036
40	3.6037	0.1122	0.2359	0.0994	0.0051
44	3.8780	0.1090	0.2430	0.1083	0.0068
48	4.2817	0.1071	0.2214	0.1078	0.0047
52	4.5503	0.0802	0.2998	0.0858	0.0037
56	4.8024	0.0993	0.3187	0.1002	0.0071
60	5.0179	0.0854	0.3225	0.0823	0.0056
64	5.2968	0.1227	0.3547	0.1181	0.0119
68	5.5242	0.0884	0.3613	0.0900	0.0059
72	5.6333	0.1090	0.3171	0.1033	0.0099
76	5.7861	0.1135	0.3537	0.1161	0.0121
80	5.9292	0.0630	0.3374	0.0630	0.0027
84	6.0984	0.1438	0.3695	0.1455	0.0202
88	6.1428	0.1120	0.3602	0.1128	0.0120
92	6.2027	0.1223	0.3700	0.1222	0.0139
96	6.1962	0.1525	0.3513	0.1481	0.0175
98	6.2168	0.1729	0.3581	0.1613	0.0177
100	6.1968	0.1972	0.3691	0.1343	0.0137
102	6.1620	0.2520	0.3571	0.1754	0.0207
104	6.2130	0.3580	0.3546	0.2467	0.0353
106	6.2164	0.4265	0.3521	0.3470	0.0607
108	6.0122	0.4175	0.3408	0.2937	0.0425
110	5.7198	0.3354	0.3342	0.2311	0.0257
112	5.5931	0.3162	0.3344	0.2463	0.0262
114	5.5333	0.2996	0.3364	0.2363	0.0257
116	5.5100	0.2992	0.3446	0.2471	0.0313
120	5.3121	0.3528	0.3436	0.2108	0.0359
124	5.1467	0.3558	0.3376	0.2371	0.0463
128	4.8953	0.3636	0.3278	0.2625	0.0578
132	4.6968	0.2877	0.2978	0.2355	0.0385
136	4.4358	0.3146	0.2652	0.2264	0.0383
140	4.1994	0.2948	0.2653	0.2266	0.0341
144	3.8511	0.2805	0.2196	0.2174	0.0288
148	3.5329	0.2688	0.1932	0.1822	0.0218

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
152	3.1972	0.2271	0.1826	0.1656	0.0121
156	2.8405	0.2531	0.1619	0.1697	0.0156
160	2.4740	0.2461	0.1686	0.1882	0.0228
164	2.0768	0.2065	0.1444	0.1491	0.0116
168	1.6524	0.1949	0.1027	0.1413	0.0086
172	1.2388	0.1797	0.0779	0.1283	0.0096
176	0.8439	0.1576	0.0453	0.1000	0.0050
180	0.4782	0.1216	0.0369	0.0616	0.0038

SPRE s/d = 44

r/R = 0.200

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.3218	0.1214	0.0325	0.0732	0.0046
8	0.5833	0.1298	0.0233	0.0869	0.0021
12	0.8923	0.1143	0.0540	0.0778	0.0005
16	1.3195	0.1234	0.0843	0.0902	0.0012
20	1.7432	0.1054	0.1108	0.0799	0.0014
24	2.1165	0.1017	0.1356	0.0731	0.0004
28	2.4421	0.1052	0.1751	0.0843	0.0009
32	2.8522	0.1049	0.1798	0.0824	0.0019
36	3.2515	0.1219	0.2000	0.0948	0.0066
40	3.5736	0.0832	0.2270	0.0709	0.0019
44	3.8768	0.0984	0.2572	0.0947	0.0056
48	4.2647	0.0957	0.2361	0.1059	0.0016
52	4.5458	0.0932	0.3048	0.0939	0.0064
56	4.7973	0.1328	0.3237	0.1424	0.0164
60	5.0154	0.1470	0.3313	0.1588	0.0215
64	5.2706	0.0861	0.3343	0.0849	0.0050
68	5.5352	0.0501	0.3746	0.0497	0.0004
72	5.6319	0.1000	0.3190	0.1039	0.0088
76	5.7749	0.0711	0.3408	0.0731	0.0042
80	5.9385	0.0685	0.3451	0.0661	0.0035
84	6.0947	0.0611	0.3558	0.0630	0.0032
88	6.1608	0.1058	0.3795	0.1090	0.0110
92	6.2119	0.1235	0.3697	0.1335	0.0147
96	6.2149	0.1487	0.3599	0.1450	0.0166
98	6.1963	0.1532	0.3512	0.1535	0.0141
100	6.2010	0.2004	0.3370	0.1685	0.0183

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
102	6.2352	0.2600	0.3406	0.1930	0.0241
104	6.4041	0.3608	0.3786	0.2667	0.0368
106	6.4804	0.3351	0.3592	0.2978	0.0343
108	6.0446	0.3795	0.3513	0.2665	0.0355
110	5.8585	0.2963	0.3440	0.2228	0.0129
112	5.7088	0.2972	0.3394	0.2234	0.0282
114	5.6487	0.2578	0.3531	0.2041	0.0101
116	5.6085	0.2823	0.3437	0.2206	0.0323
120	5.4756	0.2825	0.2949	0.2258	0.0260
124	5.2777	0.2753	0.3206	0.1828	0.0213
128	5.0670	0.2732	0.3307	0.2191	0.0275
132	4.8251	0.2567	0.3026	0.2376	0.0294
136	4.6104	0.2517	0.2683	0.1895	0.0226
140	4.2676	0.2449	0.2233	0.1848	0.0136
144	3.9821	0.2608	0.2402	0.2007	0.0157
148	3.6069	0.2522	0.1975	0.1909	0.0226
152	3.2497	0.2246	0.1864	0.1701	0.0148
156	2.9425	0.2000	0.1871	0.1963	0.0144
160	2.5362	0.2062	0.1515	0.1625	0.0096
164	2.1672	0.2132	0.1484	0.1612	0.0112
168	1.7292	0.1946	0.0860	0.1311	0.0087
172	1.3276	0.1736	0.0847	0.1218	0.0095
176	0.9160	0.1407	0.0504	0.0934	0.0022
180	0.5112	0.1200	0.0324	0.0633	0.0025

SPRE s/d = 44

r/R = 0.000

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
4	0.3839	0.1349	0.0627	0.0720	0.0057
8	0.6103	0.1261	0.0546	0.0854	0.0042
12	0.8879	0.1053	0.0674	0.0825	0.0014
16	1.2943	0.1036	0.0970	0.0940	0.0023
20	1.7137	0.1019	0.1288	0.0889	0.0031
24	2.0703	0.1117	0.1373	0.0920	0.0038
28	2.4149	0.1082	0.1388	0.0883	0.0038
32	2.8266	0.0890	0.1837	0.0845	0.0023
36	3.2417	0.1077	0.2113	0.0964	0.0065
40	3.5675	0.0928	0.2212	0.0861	0.0040

θ	$\bar{u}$	u'	$\bar{v}$	v'	$-\overline{u'v'}$
deg.	m/sec	m/sec	m/sec	m/sec	m^2/sec^2
44	3.8608	0.0822	0.2443	0.0774	0.0033
48	4.2567	0.0905	0.2126	0.0933	0.0017
52	4.5250	0.1162	0.2763	0.1145	0.0106
56	4.7839	0.0997	0.3043	0.1090	0.0082
60	5.0038	0.0617	0.3078	0.0580	0.0024
64	5.2806	0.1253	0.3439	0.1349	0.0146
68	5.5157	0.1079	0.3568	0.1236	0.0116
72	5.6445	0.1084	0.3154	0.1105	0.0110
76	5.7890	0.0822	0.3354	0.0832	0.0053
80	5.9464	0.1384	0.3315	0.1488	0.0188
84	6.0788	0.0946	0.3368	0.0953	0.0086
88	6.1579	0.1264	0.3538	0.1344	0.0159
92	6.1951	0.0967	0.3322	0.0955	0.0070
96	6.2159	0.1457	0.3407	0.1430	0.0148
98	6.2116	0.1568	0.3357	0.1380	0.0114
100	6.2340	0.1790	0.3510	0.1723	0.0127
102	6.2778	0.2720	0.3592	0.2363	0.0216
104	6.5112	0.4084	0.3821	0.3192	0.0664
106	6.5550	0.3946	0.3594	0.2485	0.0029
108	6.2652	0.3738	0.3288	0.2531	0.0285
110	5.9013	0.2955	0.2964	0.2094	0.0077
112	5.7703	0.2838	0.3259	0.2064	0.0062
114	5.7274	0.2442	0.3221	0.2104	-0.0012
116	5.6787	0.2510	0.3289	0.2202	0.0018
120	5.5319	0.2705	0.3210	0.2043	0.0054
124	5.3399	0.2302	0.3212	0.1922	0.0043
128	5.1066	0.2464	0.2975	0.2122	0.0041
132	4.8893	0.2452	0.2719	0.2035	0.0154
136	4.5900	0.2266	0.2309	0.1790	0.0125
140	4.3290	0.2341	0.2165	0.1891	0.0140
144	4.0434	0.2199	0.1938	0.1919	0.0047
148	3.6807	0.2092	0.2216	0.1866	0.0058
152	3.3379	0.2166	0.1729	0.1783	0.0068
156	2.9843	0.1988	0.1791	0.1661	0.0052
160	2.5907	0.1915	0.1396	0.1525	0.0094
164	2.1900	0.1907	0.1173	0.1423	0.0020
168	1.7311	0.1866	0.1025	0.1279	0.0021
172	1.3773	0.1572	0.0756	0.1238	0.0018
176	0.9349	0.1327	0.0420	0.1049	0.0022
180	0.5284	0.1089	0.0245	0.0662	0.0017

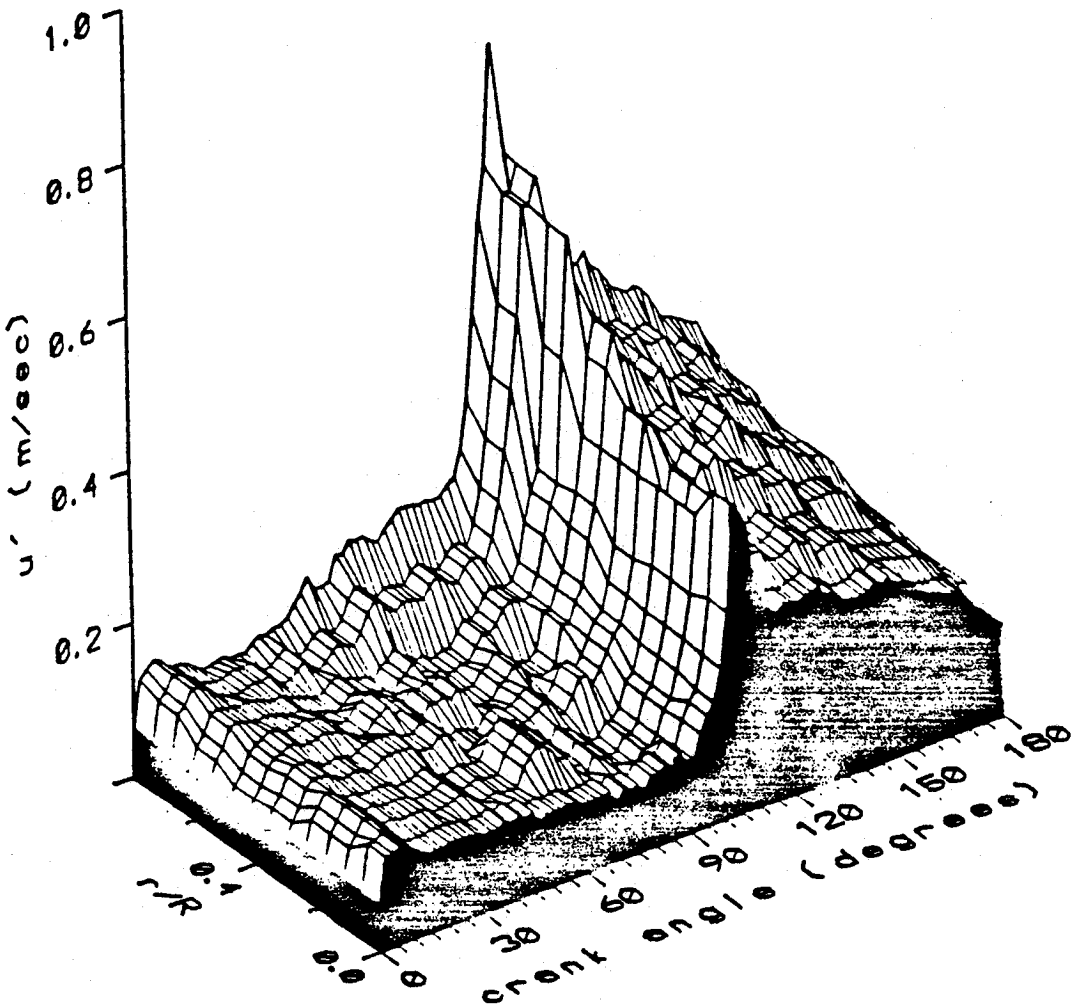


Figure 23: Streamwise velocity fluctuation at s/d = 44 (smoothed)

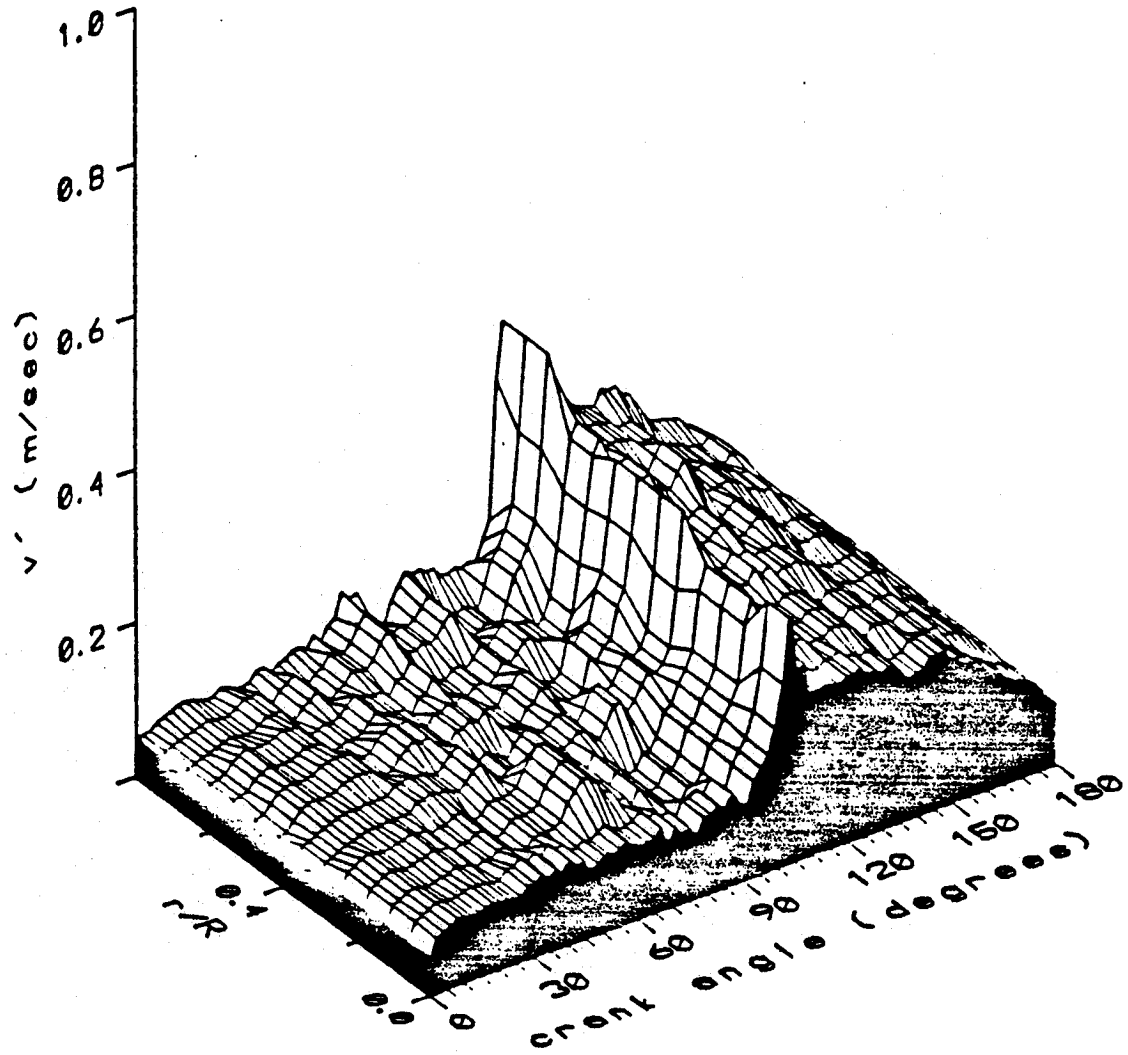


Figure 24: Radial velocity fluctuation at s/d = 44 (smoothed)

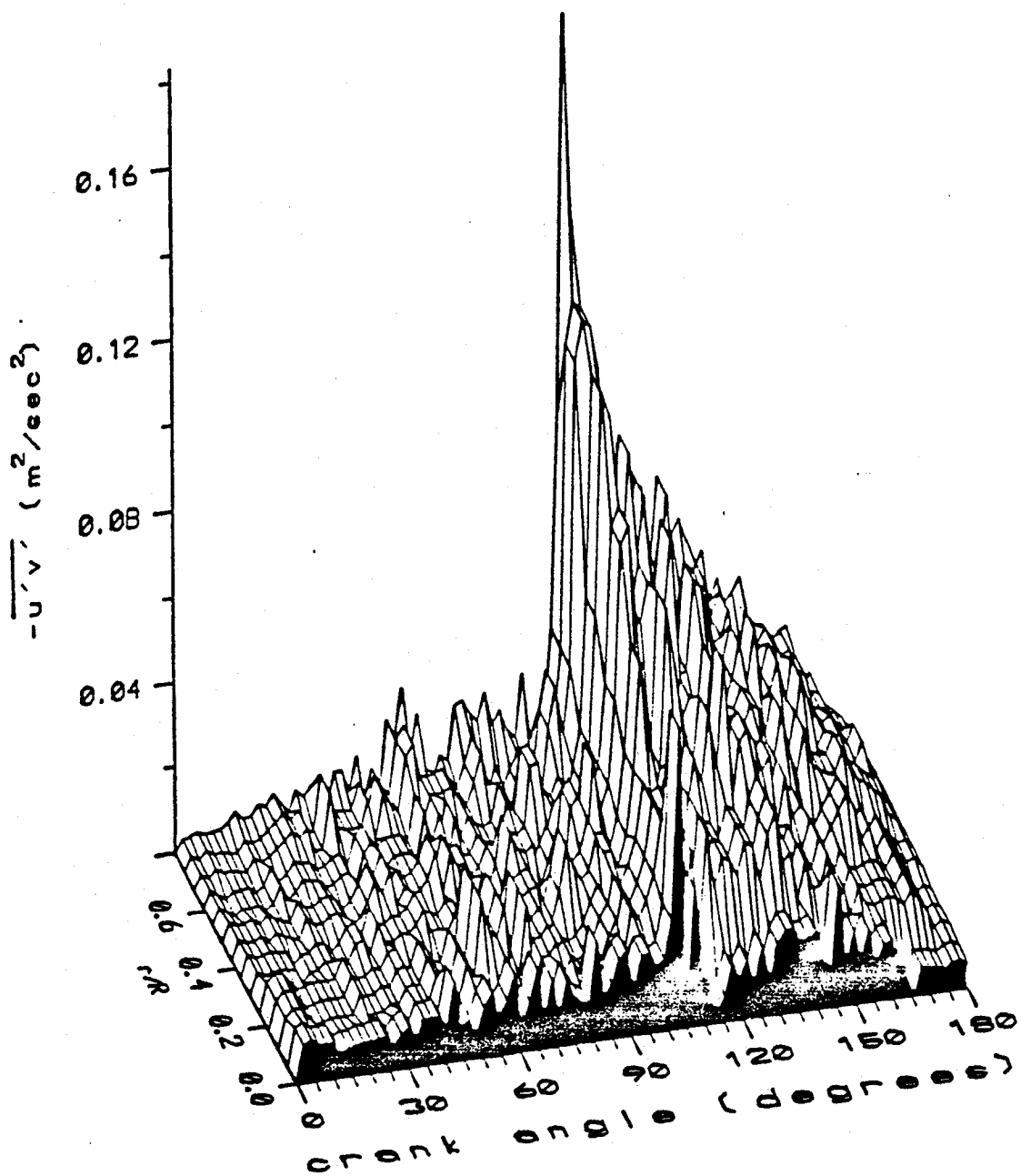


Figure 25: Reynolds shear stress at s/d = 44
(not smoothed)

I. Bulk-mean velocity, u_m, friction velocity, u_*, skin friction coefficient, c_f, and Couette flow model versus crank position, θ

Note: The symbol in the "model" column indicates which of the Couette flow models, laminar (l) or turbulent (t), was used in processing.

s/d = 0.33

θ (°)	u_m (m/sec)	u_* (m/sec)	c_f	model
20	1.518	0.138	0.01653	l
30	2.303	0.158	0.00941	t
40	3.001	0.182	0.00735	t
50	3.703	0.215	0.00674	t
60	4.130	0.235	0.00648	t
70	4.605	0.242	0.00552	l
80	4.812	0.252	0.00548	l
90	4.808	0.248	0.00532	l
100	4.746	0.242	0.00520	l
110	4.533	0.233	0.00528	l
120	4.186	0.215	0.00528	l
130	3.732	0.198	0.00563	l
140	3.164	0.167	0.00557	l
150	2.457	0.134	0.00595	l

s/d = 16

θ (°)	u_m (m/sec)	u_* (m/sec)	c_f	model
20	1.539	0.140	0.01656	l
30	2.387	0.187	0.01228	l
40	3.061	0.189	0.00762	l
50	3.701	0.198	0.00573	l
60	4.131	0.200	0.00469	l
70	4.511	0.241	0.00571	t
80	4.741	0.282	0.00708	t
90	4.806	0.294	0.00749	t
100	4.826	0.278	0.00664	t
110	4.597	0.210	0.00417	t
120	4.256	0.169	0.00315	l
130	3.791	0.139	0.00269	l
140	3.279	0.090	0.00151	l
150	2.607	0.087	0.00223	l

s/d = 30

θ (°)	u_m (m/sec)	u_* (m/sec)	c_f	model
20	1.479	0.140	0.01792	l
30	2.201	0.156	0.01005	l
40	2.878	0.175	0.00740	l
50	3.465	0.190	0.00602	l
60	3.886	0.193	0.00493	l
70	4.246	0.193	0.00413	l
80	4.441	0.189	0.00362	l
90	4.518	0.318	0.00991	t
100	4.548	0.315	0.00959	t
110	4.321	0.318	0.01083	t
120	4.080	0.290	0.01010	t
130	3.667	0.248	0.00915	t
140	3.176	0.200	0.00793	t
150	2.517	0.142	0.00637	t
160	1.849	0.100	0.00585	t
170	0.995	0.051	0.00525	t

s/d = 44

θ (°)	u_m (m/sec)	u_* (m/sec)	c_f	model
20	1.578	0.147	0.01735	l
30	2.295	0.170	0.01097	l
40	2.962	0.180	0.00739	l
50	3.569	0.189	0.00561	l
60	3.981	0.192	0.00465	l
70	4.358	0.200	0.00421	l
80	4.567	0.195	0.00365	l
90	4.631	0.188	0.00330	l
100	4.624	0.182	0.00310	l
110	4.445	0.390	0.01540	t
120	4.142	0.298	0.01035	t
130	3.708	0.267	0.01037	t
140	3.185	0.220	0.00954	t
150	2.511	0.156	0.00772	t
160	1.755	0.103	0.00689	t
170	0.891	0.040	0.00403	t

II. Processed (θ, u^+, y^+) Data: Experiment and Couette Flow Model

s/d=0.33 Experiment

20°

y^+	u^+
1.31	0.78
1.81	0.96
2.30	1.12
2.92	1.77
3.54	2.34
4.78	4.21
6.02	5.36
9.74	7.98
13.46	9.42
25.85	11.54
50.63	11.74
75.41	11.58
100.19	11.30
124.97	11.25
149.75	11.12
186.92	11.21

40°

y^+	u^+
1.73	1.65
2.39	2.30
3.04	2.71
3.86	4.03
4.67	4.68
6.31	6.80
7.94	8.34
12.84	12.04
17.75	13.97
34.09	16.56
66.77	17.05
99.45	17.16
132.13	17.01
164.81	17.03
197.50	17.11
246.52	17.07

60°

y^+	u^+
2.24	2.29
3.08	3.12
3.92	3.63
4.98	4.83
6.03	5.52
8.14	7.78
10.25	9.57
16.58	13.22
22.91	15.34
44.01	17.57
86.21	17.73
128.41	17.71
170.61	17.56
212.81	17.52
255.01	17.51
318.31	17.58

30°

y^+	u^+
1.50	1.17
2.07	1.61
2.64	1.96
3.35	3.08
4.06	3.75
5.48	5.63
6.89	7.04
11.15	10.13
15.41	12.08
29.59	14.60
57.96	15.26
86.34	15.34
114.71	15.28
143.08	15.32
171.45	15.29
214.01	15.35

50°

y^+	u^+
2.05	2.13
2.82	2.96
3.59	3.33
4.56	4.56
5.52	5.39
7.45	7.72
9.38	9.12
15.17	12.79
20.96	14.88
40.27	17.13
78.88	17.36
117.48	17.44
156.09	17.30
194.70	17.26
233.31	17.27
291.22	17.32

70°

y^+	u^+
2.30	2.63
3.17	3.44
4.04	3.94
5.13	5.19
6.21	6.16
8.39	8.48
10.56	10.33
17.08	14.45
23.60	16.69
45.32	18.77
88.78	18.81
132.24	18.77
175.69	18.60
219.15	18.54
262.60	18.51
327.79	18.57

80°

y^+	u^+
2.40	2.71
3.30	3.54
4.21	4.11
5.34	5.48
6.47	6.28
8.73	8.80
11.00	10.54
17.78	14.89
24.57	17.19
47.20	18.87
92.45	18.83
137.70	18.81
182.95	18.63
228.20	18.58
273.46	18.58
341.33	18.58

100°

y^+	u^+
2.30	2.71
3.17	3.31
4.04	3.87
5.13	5.14
6.21	6.01
8.39	8.34
10.56	10.20
17.08	14.70
23.60	17.38
45.32	19.76
88.78	19.69
132.24	19.66
175.69	19.46
219.15	19.47
262.60	19.47
327.79	19.46

120°

y^+	u^+
2.05	2.09
2.82	3.05
3.59	3.47
4.56	4.69
5.52	5.43
7.45	7.68
9.38	9.53
15.17	14.10
20.96	16.99
40.27	19.97
78.88	19.94
117.48	19.90
156.09	19.70
194.70	19.67
233.31	19.66
291.22	19.77

90°

y^+	u^+
2.36	2.66
3.25	3.38
4.14	3.96
5.25	5.26
6.37	6.07
8.59	8.43
10.82	10.35
17.50	14.66
24.18	17.28
46.45	19.38
90.98	19.31
135.52	19.28
180.05	19.09
224.58	19.02
269.12	19.02
335.92	19.04

110°

y^+	u^+
2.22	2.40
3.05	3.13
3.89	3.66
4.94	4.89
5.98	5.73
8.08	8.02
10.17	9.89
16.44	14.26
22.72	17.09
43.64	19.75
85.48	19.68
127.32	19.71
169.16	19.47
211.00	19.54
252.84	19.53
315.60	19.43

130°

y^+	u^+
1.88	1.74
2.60	2.53
3.31	3.09
4.20	4.16
5.08	4.94
6.86	7.08
8.64	8.82
13.97	13.33
19.31	16.45
37.08	19.60
72.64	19.60
108.19	19.55
143.75	19.38
179.30	19.32
214.86	19.32
268.19	19.40

s/d=0.33 Model

140°

y^+	u^+
1.59	1.33
2.19	1.80
2.79	2.27
3.54	3.70
4.29	4.29
5.79	6.39
7.29	8.19
11.79	12.69
16.28	16.09
31.28	20.05
61.27	20.17
91.25	20.10
121.24	19.90
151.23	19.77
181.22	19.78
226.20	19.89

150°

y^+	u^+
1.28	0.89
1.76	1.07
2.24	1.25
2.84	2.09
3.44	2.80
4.64	5.12
5.85	6.77
9.46	11.32
13.07	14.87
25.10	19.69
49.16	20.01
73.22	19.90
97.28	19.79
121.35	19.63
145.41	19.63
181.50	19.83

20°

y^+	u^+
1.00	0.98
2.00	1.91
3.00	2.80
4.00	3.65
5.00	4.45
6.00	5.21
7.00	5.92
8.00	6.59
9.00	7.21
10.00	7.79
11.00	8.33
12.00	8.82
13.00	9.27
14.00	9.68
15.00	10.04

30°

y^+	u^+
1.0	1.00
2.0	2.00
3.0	3.00
4.0	4.00
5.0	4.99
6.0	5.99
7.0	6.97
8.0	7.95
9.0	8.91
10.0	9.85
11.0	10.76
12.0	11.65
13.0	12.50
14.0	13.32
15.0	14.10
16.0	14.85
17.0	15.55
18.0	16.22
19.0	16.86
20.0	17.45
21.0	18.02
22.0	18.55
23.0	19.06
24.0	19.53
25.0	19.98
26.0	20.41
27.0	20.82
28.0	21.20
29.0	21.56
30.0	21.91
31.0	22.24
32.0	22.55
33.0	22.85
34.0	23.13
35.0	23.40
36.0	23.66
37.0	23.91
38.0	24.15
39.0	24.38
40.0	24.59
41.0	24.80
42.0	25.00
43.0	25.20
44.0	25.38
45.0	25.56
46.0	25.74
47.0	25.90
48.0	26.06
49.0	26.22
50.0	26.37

40°		50°		60°	
y^+	u^+	y^+	u^+	y^+	u^+
1.0	1.00	1.0	1.00	1.0	1.00
2.0	2.00	2.0	2.00	2.0	2.00
3.0	3.00	3.0	2.99	3.0	2.99
4.0	3.99	4.0	3.97	4.0	3.97
5.0	4.96	5.0	4.92	5.0	4.90
6.0	5.90	6.0	5.82	6.0	5.79
7.0	6.81	7.0	6.66	7.0	6.60
8.0	7.66	8.0	7.43	8.0	7.34
9.0	8.46	9.0	8.13	9.0	8.00
10.0	9.19	10.0	8.75	10.0	8.59
11.0	9.86	11.0	9.32	11.0	9.12
12.0	10.47	12.0	9.82	12.0	9.59
13.0	11.03	13.0	10.28	13.0	10.01
14.0	11.54	14.0	10.69	14.0	10.40
15.0	12.01	15.0	11.06	15.0	10.75
16.0	12.44	16.0	11.40	16.0	11.06
17.0	12.83	17.0	11.72	17.0	11.35
18.0	13.19	18.0	12.00	18.0	11.62
19.0	13.52	19.0	12.27	19.0	11.86
20.0	13.83	20.0	12.51	20.0	12.09
21.0	14.12	21.0	12.73	21.0	12.30
22.0	14.39	22.0	12.95	22.0	12.49
23.0	14.63	23.0	13.14	23.0	12.67
24.0	14.87	24.0	13.33	24.0	12.84
25.0	15.09	25.0	13.50	25.0	13.00
26.0	15.29	26.0	13.66	26.0	13.16
27.0	15.48	27.0	13.81	27.0	13.30
28.0	15.67	28.0	13.96	28.0	13.43
29.0	15.84	29.0	14.10	29.0	13.56
30.0	16.00	30.0	14.23	30.0	13.68
31.0	16.16	31.0	14.35	31.0	13.80
32.0	16.31	32.0	14.47	32.0	13.91
33.0	16.45	33.0	14.58	33.0	14.02
34.0	16.58	34.0	14.69	34.0	14.12
35.0	16.71	35.0	14.79	35.0	14.21
36.0	16.83	36.0	14.89	36.0	14.31
37.0	16.95	37.0	14.99	37.0	14.40
38.0	17.06	38.0	15.08	38.0	14.48
39.0	17.17	39.0	15.16	39.0	14.57
40.0	17.27	40.0	15.25	40.0	14.65
41.0	17.37	41.0	15.33	41.0	14.72
42.0	17.47	42.0	15.41	42.0	14.80
43.0	17.56	43.0	15.49	43.0	14.87
44.0	17.65	44.0	15.56	44.0	14.94
45.0	17.73	45.0	15.63	45.0	15.01
46.0	17.82	46.0	15.70	46.0	15.07
47.0	17.90	47.0	15.77	47.0	15.14
48.0	17.97	48.0	15.83	48.0	15.20
49.0	18.05	49.0	15.89	49.0	15.26
50.0	18.12	50.0	15.96	50.0	15.32

70°

y^+	u^+
1.00	1.00
2.00	1.99
3.00	2.99
4.00	3.98
5.00	4.96
6.00	5.95
7.00	6.93
8.00	7.90
9.00	8.88
10.00	9.85
11.00	10.82
12.00	11.79
13.00	12.75
14.00	13.71
15.00	14.66

90°

y^+	u^+
1.00	1.00
2.00	2.00
3.00	3.00
4.00	4.00
5.00	5.00
6.00	6.00
7.00	7.00
8.00	8.00
9.00	9.00
10.00	10.00
11.00	11.00
12.00	12.00
13.00	13.00
14.00	14.00
15.00	15.00

110°

y^+	u^+
1.00	1.00
2.00	2.01
3.00	3.02
4.00	4.03
5.00	5.04
6.00	6.06
7.00	7.08
8.00	8.11
9.00	9.14
10.00	10.17
11.00	11.20
12.00	12.24
13.00	13.28
14.00	14.33
15.00	15.38

80°

y^+	u^+
1.00	1.00
2.00	2.00
3.00	2.99
4.00	3.99
5.00	4.98
6.00	5.98
7.00	6.97
8.00	7.96
9.00	8.95
10.00	9.93
11.00	10.92
12.00	11.90
13.00	12.89
14.00	13.87
15.00	14.85

100°

y^+	u^+
1.00	1.00
2.00	2.00
3.00	3.01
4.00	4.01
5.00	5.02
6.00	6.03
7.00	7.04
8.00	8.05
9.00	9.06
10.00	10.08
11.00	11.09
12.00	12.11
13.00	13.13
14.00	14.15
15.00	15.17

120°

y^+	u^+
1.00	1.00
2.00	2.01
3.00	3.03
4.00	4.05
5.00	5.08
6.00	6.11
7.00	7.15
8.00	8.20
9.00	9.25
10.00	10.31
11.00	11.38
12.00	12.45
13.00	13.52
14.00	14.61
15.00	15.70

130°

y^+	u^+
1.00	1.01
2.00	2.02
3.00	3.05
4.00	4.08
5.00	5.13
6.00	6.18
7.00	7.25
8.00	8.33
9.00	9.41
10.00	10.51
11.00	11.62
12.00	12.74
13.00	13.86
14.00	15.00
15.00	16.15

140°

y^+	u^+
1.00	1.01
2.00	2.04
3.00	3.09
4.00	4.16
5.00	5.25
6.00	6.37
7.00	7.50
8.00	8.65
9.00	9.82
10.00	11.01
11.00	12.23
12.00	13.46
13.00	14.72
14.00	15.99
15.00	17.28

150°

y^+	u^+
1.00	1.02
2.00	2.09
3.00	3.20
4.00	4.36
5.00	5.56
6.00	6.80
7.00	8.09
8.00	9.42
9.00	10.80
10.00	12.22
11.00	13.69
12.00	15.20
13.00	16.75
14.00	18.35
15.00	20.00

20°

y^+	u^+
0.95	0.81
1.47	0.91
3.28	2.41
7.92	7.02
11.79	9.11
24.69	12.03
50.49	12.48
76.29	12.24
102.09	11.87
127.90	11.49
153.70	11.26
192.40	11.37

30°

y^+	u^+
1.27	1.30
1.96	1.76
4.38	4.22
10.58	9.35
15.75	11.60
32.98	13.82
67.44	13.87
101.91	13.68
136.37	13.50
170.83	13.60
205.29	13.77
256.99	14.12

40°

y^+	u^+
1.29	1.21
1.99	1.60
4.42	3.98
10.69	9.12
15.92	12.04
33.33	16.91
68.16	18.44
103.00	18.52
137.83	18.32
172.66	17.66
207.49	17.00
259.74	16.88

50°

y^+	u^+
1.35	1.34
2.08	1.82
4.63	4.26
11.20	9.62
16.68	12.73
34.92	18.72
71.41	21.37
107.90	21.65
144.39	21.58
180.88	21.38
217.37	20.76
272.11	19.44

60°

y^+	u^+
1.36	1.44
2.10	1.95
4.68	4.46
11.32	10.13
16.84	13.50
35.27	20.56
72.13	23.62
108.99	24.05
145.85	23.98
182.71	23.88
219.57	23.40
274.85	22.19

70°

y^+	u^+
1.64	1.78
2.53	2.42
5.64	5.03
13.64	10.61
20.30	13.71
42.51	18.50
86.92	20.86
131.33	21.46
175.75	21.55
220.16	21.50
264.58	21.10
331.20	20.25

80°

y^+	u^+
1.92	2.33
2.96	3.04
6.60	5.92
15.95	11.05
23.75	13.38
49.74	16.61
101.71	18.33
153.68	18.88
205.65	19.03
257.62	19.06
309.59	18.84
387.55	18.34

90°

y^+	u^+
2.00	2.46
3.09	3.21
6.88	6.18
16.63	10.92
24.76	13.04
51.85	16.11
106.03	17.57
160.22	18.39
214.40	18.56
268.58	18.68
322.76	18.39
404.04	17.94

100°

y^+	u^+
1.90	2.36
2.92	3.06
6.51	5.78
15.73	11.22
23.41	13.30
49.03	16.77
100.26	18.89
151.50	19.70
202.73	19.95
253.96	20.03
305.20	19.92
382.05	19.36

110°

y^+	u^+
1.43	1.59
2.21	2.17
4.91	4.62
11.88	10.44
17.69	13.96
37.04	20.91
75.74	24.86
114.44	25.75
153.14	25.83
191.84	25.86
230.55	25.73
288.60	25.20

120°

y^+	u^+
1.15	1.13
1.78	1.46
3.96	3.81
9.56	9.61
14.23	13.60
29.81	22.55
60.95	29.40
92.10	30.79
123.24	30.71
154.39	30.65
185.53	30.50
232.25	29.83

130°

y^+	u^+
0.95	0.94
1.46	1.09
3.25	2.99
7.86	9.16
11.71	13.31
24.52	23.61
50.13	32.26
75.75	33.87
101.37	33.81
126.98	33.79
152.60	33.72
191.02	33.56

s/d=16 Model

140°

y^+	u^+
0.61	0.90
0.95	0.88
2.11	2.27
5.09	10.54
7.58	16.03
15.87	30.86
32.46	43.41
49.05	45.92
65.63	45.93
82.22	45.89
98.81	45.75
123.68	45.33

150°

y^+	u^+
0.59	0.53
0.91	0.39
2.04	0.67
4.92	7.02
7.33	11.49
15.34	24.51
31.38	35.58
47.41	38.66
63.44	38.67
79.48	38.66
95.51	38.58
119.56	38.93

20°

y^+	u^+
1.00	0.98
2.00	1.92
3.00	2.81
4.00	3.67
5.00	4.49
6.00	5.26
7.00	5.99
8.00	6.68
9.00	7.33
10.00	7.94
11.00	8.51
12.00	9.04
13.00	9.52
14.00	9.96
15.00	10.37

30°

y^+	u^+
1.00	0.99
2.00	1.97
3.00	2.93
4.00	3.87
5.00	4.80
6.00	5.71
7.00	6.61
8.00	7.49
9.00	8.36
10.00	9.20
11.00	10.04
12.00	10.85
13.00	11.65
14.00	12.44
15.00	13.21

40°

y^+	u^+
1.00	0.99
2.00	1.97
3.00	2.94
4.00	3.89
5.00	4.83
6.00	5.75
7.00	6.67
8.00	7.56
9.00	8.45
10.00	9.32
11.00	10.17
12.00	11.02
13.00	11.85
14.00	12.66
15.00	13.46

50°

y^+	u^+
1.00	1.00
2.00	1.98
3.00	2.96
4.00	3.92
5.00	4.88
6.00	5.82
7.00	6.76
8.00	7.68
9.00	8.60
10.00	9.50
11.00	10.40
12.00	11.28
13.00	12.16
14.00	13.02
15.00	13.88

60°		70°		80°	
y^+	u^+	y^+	u^+	y^+	u^+
1.00	1.00	1.0	1.00	1.0	1.00
2.00	1.98	2.0	2.00	2.0	2.00
3.00	2.97	3.0	2.99	3.0	2.99
4.00	3.94	4.0	3.96	4.0	3.96
5.00	4.91	5.0	4.89	5.0	4.88
6.00	5.86	6.0	5.76	6.0	5.74
7.00	6.82	7.0	6.56	7.0	6.52
8.00	7.76	8.0	7.28	8.0	7.23
9.00	8.70	9.0	7.93	9.0	7.85
10.00	9.62	10.0	8.50	10.0	8.41
11.00	10.55	11.0	9.01	11.0	8.90
12.00	11.46	12.0	9.47	12.0	9.34
13.00	12.36	13.0	9.88	13.0	9.73
14.00	13.26	14.0	10.25	14.0	10.09
15.00	14.15	15.0	10.58	15.0	10.41
		16.0	10.89	16.0	10.70
		17.0	11.16	17.0	10.97
		18.0	11.42	18.0	11.22
		19.0	11.65	19.0	11.44
		20.0	11.87	20.0	11.65
		21.0	12.07	21.0	11.85
		22.0	12.26	22.0	12.03
		23.0	12.44	23.0	12.20
		24.0	12.60	24.0	12.36
		25.0	12.76	25.0	12.51
		26.0	12.90	26.0	12.65
		27.0	13.04	27.0	12.78
		28.0	13.17	28.0	12.91
		29.0	13.30	29.0	13.03
		30.0	13.41	30.0	13.14
		31.0	13.53	31.0	13.25
		32.0	13.63	32.0	13.35
		33.0	13.74	33.0	13.45
		34.0	13.83	34.0	13.55
		35.0	13.93	35.0	13.64
		36.0	14.02	36.0	13.73
		37.0	14.11	37.0	13.81
		38.0	14.19	38.0	13.89
		39.0	14.27	39.0	13.97
		40.0	14.35	40.0	14.05
		41.0	14.42	41.0	14.12
		42.0	14.49	42.0	14.19
		43.0	14.57	43.0	14.26
		44.0	14.63	44.0	14.33
		45.0	14.70	45.0	14.39
		46.0	14.76	46.0	14.45
		47.0	14.83	47.0	14.51
		48.0	14.89	48.0	14.57
		49.0	14.94	49.0	14.63
		50.0	15.00	50.0	14.69

90°		100°		110°	
y^+	u^+	y^+	u^+	y^+	u^+
1.0	1.00	1.0	1.00	1.0	1.00
2.0	2.00	2.0	2.00	2.0	2.00
3.0	2.99	3.0	2.99	3.0	2.99
4.0	3.95	4.0	3.95	4.0	3.95
5.0	4.88	5.0	4.87	5.0	4.86
6.0	5.73	6.0	5.72	6.0	5.69
7.0	6.51	7.0	6.49	7.0	6.45
8.0	7.20	8.0	7.18	8.0	7.12
9.0	7.82	9.0	7.79	9.0	7.71
10.0	8.36	10.0	8.33	10.0	8.23
11.0	8.85	11.0	8.82	11.0	8.70
12.0	9.28	12.0	9.24	12.0	9.11
13.0	9.67	13.0	9.63	13.0	9.48
14.0	10.02	14.0	9.97	14.0	9.81
15.0	10.34	15.0	10.29	15.0	10.11
16.0	10.62	16.0	10.57	16.0	10.38
17.0	10.89	17.0	10.83	17.0	10.63
18.0	11.13	18.0	11.07	18.0	10.86
19.0	11.35	19.0	11.29	19.0	11.07
20.0	11.56	20.0	11.49	20.0	11.27
21.0	11.75	21.0	11.68	21.0	11.45
22.0	11.93	22.0	11.86	22.0	11.62
23.0	12.09	23.0	12.02	23.0	11.78
24.0	12.25	24.0	12.18	24.0	11.93
25.0	12.40	25.0	12.32	25.0	12.07
26.0	12.54	26.0	12.46	26.0	12.20
27.0	12.67	27.0	12.59	27.0	12.33
28.0	12.79	28.0	12.71	28.0	12.45
29.0	12.91	29.0	12.83	29.0	12.56
30.0	13.02	30.0	12.94	30.0	12.67
31.0	13.13	31.0	13.05	31.0	12.77
32.0	13.23	32.0	13.15	32.0	12.87
33.0	13.33	33.0	13.25	33.0	12.96
34.0	13.42	34.0	13.34	34.0	13.05
35.0	13.51	35.0	13.43	35.0	13.14
36.0	13.60	36.0	13.51	36.0	13.22
37.0	13.68	37.0	13.60	37.0	13.30
38.0	13.76	38.0	13.68	38.0	13.38
39.0	13.84	39.0	13.75	39.0	13.46
40.0	13.92	40.0	13.83	40.0	13.53
41.0	13.99	41.0	13.90	41.0	13.60
42.0	14.06	42.0	13.97	42.0	13.67
43.0	14.13	43.0	14.04	43.0	13.73
44.0	14.19	44.0	14.10	44.0	13.80
45.0	14.26	45.0	14.16	45.0	13.86
46.0	14.32	46.0	14.23	46.0	13.92
47.0	14.38	47.0	14.29	47.0	13.98
48.0	14.44	48.0	14.34	48.0	14.03
49.0	14.49	49.0	14.40	49.0	14.09
50.0	14.55	50.0	14.46	50.0	14.14

s/d=30 Experiment

120°

y^+	u^+
1.00	1.01
2.00	2.02
3.00	3.06
4.00	4.10
5.00	5.16
6.00	6.22
7.00	7.31
8.00	8.40
9.00	9.50
10.00	10.62
11.00	11.75
12.00	12.90
13.00	14.05
14.00	15.22
15.00	16.40

140°

y^+	u^+
1.00	1.06
2.00	2.25
3.00	3.57
4.00	5.01
5.00	6.58
6.00	8.27
7.00	10.10
8.00	12.04
9.00	14.12
10.00	16.32
11.00	18.64
12.00	21.10
13.00	23.68
14.00	26.38
15.00	29.22

20°

y^+	u^+
0.93	0.78
1.70	1.04
2.73	1.82
3.24	2.52
4.53	4.26
5.81	5.34
9.67	7.85
13.53	9.49
26.38	11.60
52.10	11.88
77.81	11.63
103.52	11.35
129.24	11.08
154.95	11.03
193.52	10.87

130°

y^+	u^+
1.00	1.01
2.00	2.06
3.00	3.13
4.00	4.23
5.00	5.36
6.00	6.52
7.00	7.71
8.00	8.92
9.00	10.17
10.00	11.44
11.00	12.74
12.00	14.07
13.00	15.43
14.00	16.82
15.00	18.24

150°

y^+	u^+
1.00	1.08
2.00	2.32
3.00	3.71
4.00	5.27
5.00	6.98
6.00	8.85
7.00	10.87
8.00	13.06
9.00	15.40
10.00	17.91
11.00	20.57
12.00	23.39
13.00	26.36
14.00	29.50
15.00	32.79

30°

y^+	u^+
1.03	1.03
1.89	1.50
3.04	2.72
3.61	3.69
5.04	5.15
6.48	6.41
10.77	9.45
15.07	11.53
29.40	14.97
58.05	15.96
86.70	15.81
115.36	15.69
144.01	15.42
172.66	15.44
215.64	15.36

40°

y^+	u^+
1.16	1.14
2.12	1.83
3.41	3.31
4.05	4.01
5.66	5.52
7.26	6.85
12.09	10.15
16.91	12.60
32.98	16.90
65.12	18.67
97.26	18.77
129.40	18.74
161.55	18.54
193.69	18.56
241.90	18.49

60°

y^+	u^+
1.28	1.32
2.34	2.20
3.76	3.64
4.47	4.34
6.24	5.93
8.01	7.38
13.33	10.98
18.65	13.73
36.37	19.50
71.82	23.08
107.27	23.72
142.72	23.85
178.16	23.72
213.61	23.80
266.78	23.79

80°

y^+	u^+
1.84	1.30
2.88	2.14
4.27	3.63
4.96	4.30
6.70	5.92
8.44	7.36
13.64	11.10
18.85	14.24
36.21	21.72
70.92	27.25
105.63	28.36
140.35	28.76
175.06	28.68
209.77	28.77
261.84	28.80

50°

y^+	u^+
1.26	1.29
2.30	2.13
3.70	3.59
4.40	4.26
6.14	5.81
7.89	7.22
13.12	10.60
18.36	13.22
35.80	18.22
70.70	20.82
105.60	21.10
140.50	21.14
175.39	21.01
210.29	21.05
262.64	21.00

70°

y^+	u^+
1.28	1.34
2.34	2.22
3.76	3.67
4.47	4.39
6.24	6.00
8.01	7.48
13.33	11.15
18.65	14.13
36.37	20.62
71.82	25.35
107.27	26.30
142.72	26.61
178.16	26.45
213.61	26.58
266.78	26.65

90°

y^+	u^+
2.10	2.45
3.86	3.82
6.19	5.11
7.36	5.85
10.28	7.40
13.20	8.38
21.96	10.34
30.72	11.28
59.93	12.99
118.33	14.63
176.74	15.92
235.15	16.90
293.55	17.35
351.96	17.96
439.57	18.20

100°

y^+	u^+
2.08	2.47
3.82	3.74
6.13	5.18
7.29	6.15
10.18	7.56
13.08	8.90
21.75	11.06
30.43	12.20
59.36	13.63
117.22	15.04
175.07	16.01
232.93	16.75
290.78	17.08
348.64	17.36
435.42	17.46

120°

y^+	u^+
1.92	2.23
3.52	3.35
5.65	4.65
6.71	5.68
9.37	7.19
12.04	8.26
20.03	10.39
28.02	11.51
54.65	13.23
107.91	14.55
161.18	15.65
214.44	16.53
267.71	16.99
320.97	17.44
400.87	17.51

140°

y^+	u^+
1.32	1.65
2.42	2.51
3.89	3.86
4.63	4.60
6.47	6.24
8.30	7.67
13.81	10.16
19.32	11.75
37.69	14.35
74.42	16.72
111.16	18.54
147.89	19.51
184.62	19.82
221.36	19.99
276.46	19.56

110°

y^+	u^+
2.10	2.29
3.86	3.46
6.19	4.90
7.36	5.55
10.28	7.07
13.20	8.38
21.96	10.28
30.72	11.20
59.93	12.91
118.33	14.19
176.74	15.02
235.15	15.71
293.55	16.17
351.96	16.47
439.57	16.76

130°

y^+	u^+
1.64	1.93
3.01	3.12
4.83	4.48
5.74	5.44
8.02	6.94
10.29	8.19
17.13	10.62
23.96	11.83
46.73	13.73
92.28	15.37
137.83	16.53
183.39	17.64
228.93	18.10
274.49	18.50
342.81	18.44

150°

y^+	u^+
0.94	1.04
1.72	1.56
2.76	2.75
3.29	3.41
4.59	4.61
5.89	6.20
9.81	8.97
13.72	10.55
26.76	14.75
52.84	19.06
78.92	21.75
105.00	23.19
131.08	23.54
157.16	23.81
196.29	22.98

s/d=30 Model

20°

y^+	u^+
1.00	0.98
2.00	1.92
3.00	2.81
4.00	3.67
5.00	4.48
6.00	5.26
7.00	5.99
8.00	6.68
9.00	7.33
10.00	7.93
11.00	8.50
12.00	9.02
13.00	9.51
14.00	9.95
15.00	10.35

40°

y^+	u^+
1.00	0.99
2.00	1.97
3.00	2.92
4.00	3.86
5.00	4.78
6.00	5.69
7.00	6.58
8.00	7.45
9.00	8.30
10.00	9.14
11.00	9.96
12.00	10.76
13.00	11.54
14.00	12.31
15.00	13.06

60°

y^+	u^+
1.00	1.00
2.00	1.98
3.00	2.96
4.00	3.93
5.00	4.90
6.00	5.85
7.00	6.79
8.00	7.73
9.00	8.66
10.00	9.58
11.00	10.49
12.00	11.40
13.00	12.29
14.00	13.18
15.00	14.06

30°

y^+	u^+
1.00	0.99
2.00	1.94
3.00	2.88
4.00	3.78
5.00	4.66
6.00	5.50
7.00	6.33
8.00	7.12
9.00	7.89
10.00	8.62
11.00	9.33
12.00	10.02
13.00	10.67
14.00	11.30
15.00	11.90

50°

y^+	u^+
1.00	0.99
2.00	1.98
3.00	2.95
4.00	3.91
5.00	4.86
6.00	5.80
7.00	6.72
8.00	7.64
9.00	8.54
10.00	9.43
11.00	10.32
12.00	11.19
13.00	12.04
14.00	12.89
15.00	13.73

70°

y^+	u^+
1.00	1.00
2.00	1.99
3.00	2.97
4.00	3.95
5.00	4.93
6.00	5.90
7.00	6.86
8.00	7.82
9.00	8.77
10.00	9.71
11.00	10.65
12.00	11.59
13.00	12.51
14.00	13.44
15.00	14.35

80°		90°		100°	
y^+	u^+	y^+	u^+	y^+	u^+
1.00	1.00	1.0	1.00	1.0	1.00
2.00	1.99	2.0	2.00	2.0	2.00
3.00	2.99	3.0	2.99	3.0	2.99
4.00	3.98	4.0	3.95	4.0	3.95
5.00	4.96	5.0	4.88	5.0	4.87
6.00	5.94	6.0	5.73	6.0	5.72
7.00	6.92	7.0	6.51	7.0	6.50
8.00	7.90	8.0	7.20	8.0	7.19
9.00	8.87	9.0	7.82	9.0	7.80
10.00	9.84	10.0	8.36	10.0	8.34
11.00	10.81	11.0	8.85	11.0	8.83
12.00	11.78	12.0	9.28	12.0	9.26
13.00	12.74	13.0	9.67	13.0	9.64
14.00	13.70	14.0	10.02	14.0	9.99
15.00	14.65	15.0	10.34	15.0	10.30
		16.0	10.62	16.0	10.59
		17.0	10.89	17.0	10.85
		18.0	11.13	18.0	11.09
		19.0	11.35	19.0	11.31
		20.0	11.56	20.0	11.51
		21.0	11.75	21.0	11.70
		22.0	11.93	22.0	11.88
		23.0	12.09	23.0	12.04
		24.0	12.25	24.0	12.20
		25.0	12.40	25.0	12.35
		26.0	12.54	26.0	12.48
		27.0	12.67	27.0	12.61
		28.0	12.79	28.0	12.74
		29.0	12.91	29.0	12.85
		30.0	13.02	30.0	12.97
		31.0	13.13	31.0	13.07
		32.0	13.23	32.0	13.17
		33.0	13.33	33.0	13.27
		34.0	13.42	34.0	13.37
		35.0	13.51	35.0	13.45
		36.0	13.60	36.0	13.54
		37.0	13.68	37.0	13.62
		38.0	13.76	38.0	13.70
		39.0	13.84	39.0	13.78
		40.0	13.92	40.0	13.85
		41.0	13.99	41.0	13.93
		42.0	14.06	42.0	14.00
		43.0	14.13	43.0	14.06
		44.0	14.19	44.0	14.13
		45.0	14.26	45.0	14.19
		46.0	14.32	46.0	14.25
		47.0	14.38	47.0	14.31
		48.0	14.44	48.0	14.37
		49.0	14.49	49.0	14.43
		50.0	14.55	50.0	14.48

110°		120°		130°	
y^+	u^+	y^+	u^+	y^+	u^+
1.0	1.00	1.0	1.00	1.0	1.00
2.0	2.00	2.0	2.00	2.0	2.00
3.0	2.99	3.0	2.99	3.0	2.99
4.0	3.95	4.0	3.95	4.0	3.95
5.0	4.87	5.0	4.86	5.0	4.85
6.0	5.72	6.0	5.71	6.0	5.69
7.0	6.49	7.0	6.47	7.0	6.44
8.0	7.18	8.0	7.15	8.0	7.11
9.0	7.78	9.0	7.76	9.0	7.69
10.0	8.32	10.0	8.29	10.0	8.21
11.0	8.80	11.0	8.76	11.0	8.67
12.0	9.23	12.0	9.18	12.0	9.08
13.0	9.61	13.0	9.56	13.0	9.45
14.0	9.96	14.0	9.90	14.0	9.78
15.0	10.27	15.0	10.21	15.0	10.08
16.0	10.55	16.0	10.49	16.0	10.35
17.0	10.81	17.0	10.74	17.0	10.60
18.0	11.05	18.0	10.98	18.0	10.82
19.0	11.27	19.0	11.19	19.0	11.03
20.0	11.47	20.0	11.39	20.0	11.23
21.0	11.66	21.0	11.58	21.0	11.41
22.0	11.83	22.0	11.75	22.0	11.58
23.0	12.00	23.0	11.91	23.0	11.74
24.0	12.15	24.0	12.07	24.0	11.88
25.0	12.30	25.0	12.21	25.0	12.02
26.0	12.44	26.0	12.34	26.0	12.16
27.0	12.56	27.0	12.47	27.0	12.28
28.0	12.69	28.0	12.59	28.0	12.40
29.0	12.80	29.0	12.71	29.0	12.51
30.0	12.92	30.0	12.82	30.0	12.62
31.0	13.02	31.0	12.92	31.0	12.72
32.0	13.12	32.0	13.02	32.0	12.82
33.0	13.22	33.0	13.12	33.0	12.91
34.0	13.31	34.0	13.21	34.0	13.00
35.0	13.40	35.0	13.30	35.0	13.09
36.0	13.49	36.0	13.38	36.0	13.17
37.0	13.57	37.0	13.47	37.0	13.25
38.0	13.65	38.0	13.54	38.0	13.33
39.0	13.73	39.0	13.62	39.0	13.40
40.0	13.80	40.0	13.69	40.0	13.48
41.0	13.87	41.0	13.77	41.0	13.55
42.0	13.94	42.0	13.83	42.0	13.61
43.0	14.01	43.0	13.90	43.0	13.68
44.0	14.07	44.0	13.97	44.0	13.74
45.0	14.14	45.0	14.03	45.0	13.80
46.0	14.20	46.0	14.09	46.0	13.86
47.0	14.26	47.0	14.15	47.0	13.92
48.0	14.31	48.0	14.21	48.0	13.98
49.0	14.37	49.0	14.26	49.0	14.03
50.0	14.43	50.0	14.32	50.0	14.09

140°

y^+	u^+
1.0	1.00
2.0	2.00
3.0	2.98
4.0	3.93
5.0	4.83
6.0	5.64
7.0	6.36
8.0	6.99
9.0	7.55
10.0	8.04
11.0	8.47
12.0	8.85
13.0	9.20
14.0	9.50
15.0	9.78
16.0	10.04
17.0	10.27
18.0	10.48
19.0	10.68
20.0	10.86
21.0	11.03
22.0	11.19
23.0	11.34
24.0	11.48
25.0	11.61
26.0	11.74
27.0	11.86
28.0	11.97
29.0	12.07
30.0	12.18
31.0	12.27
32.0	12.37
33.0	12.46
34.0	12.54
35.0	12.63
36.0	12.71
37.0	12.78
38.0	12.86
39.0	12.93
40.0	13.00
41.0	13.07
42.0	13.13
43.0	13.19
44.0	13.25
45.0	13.31
46.0	13.37
47.0	13.43
48.0	13.48
49.0	13.54
50.0	13.59

150°

y^+	u^+
1.0	1.00
2.0	2.00
3.0	2.97
4.0	3.88
5.0	4.71
6.0	5.44
7.0	6.07
8.0	6.61
9.0	7.07
10.0	7.48
11.0	7.84
12.0	8.16
13.0	8.44
14.0	8.70
15.0	8.93
16.0	9.14
17.0	9.33
18.0	9.51
19.0	9.68
20.0	9.83
21.0	9.98
22.0	10.11
23.0	10.24
24.0	10.36
25.0	10.47
26.0	10.58
27.0	10.69
28.0	10.78
29.0	10.88
30.0	10.97
31.0	11.05
32.0	11.14
33.0	11.22
34.0	11.30
35.0	11.37
36.0	11.44
37.0	11.51
38.0	11.58
39.0	11.65
40.0	11.71
41.0	11.77
42.0	11.83
43.0	11.89
44.0	11.95
45.0	12.00
46.0	12.06
47.0	12.11
48.0	12.16
49.0	12.21
50.0	12.26

s/d=44 Experiment

20°

y^+	u^+
0.16	0.97
0.43	0.83
0.98	0.84
1.53	1.10
2.07	1.21
3.44	2.96
4.81	4.68
13.01	9.20
26.69	11.63
54.04	12.17
81.39	11.94
108.74	11.59
136.09	11.08
163.44	10.98
204.46	11.31

30°

y^+	u^+
0.18	0.98
0.50	0.91
1.13	1.02
1.76	1.37
2.40	1.59
3.98	3.74
5.56	5.17
15.05	10.48
30.86	14.23
62.49	15.15
94.12	15.21
125.75	15.08
157.38	14.79
189.01	14.76
236.45	14.88

40°

y^+	u^+
0.19	1.05
0.53	1.00
1.20	1.21
1.87	1.68
2.54	1.93
4.21	4.11
5.89	5.71
15.93	11.74
32.68	16.73
66.17	18.68
99.66	18.79
133.15	18.79
166.64	18.61
200.13	18.63
250.36	18.87

60°

y^+	u^+
0.21	1.16
0.56	1.15
1.28	1.46
1.99	2.06
2.71	2.39
4.49	4.57
6.28	6.26
17.00	13.01
34.86	19.83
70.58	23.80
106.30	24.43
142.02	24.59
177.75	24.39
213.47	24.42
267.05	24.83

80°

y^+	u^+
0.21	1.15
0.57	1.15
1.30	1.46
2.02	2.04
2.75	2.42
4.56	4.54
6.38	6.22
17.26	13.33
35.40	21.07
71.68	27.06
107.96	28.32
144.24	28.59
180.52	28.51
216.80	28.58
271.22	29.04

50°

y^+	u^+
0.20	1.15
0.56	1.14
1.26	1.45
1.96	2.03
2.67	2.35
4.42	4.57
6.18	6.21
16.73	12.78
34.31	18.55
69.48	21.58
104.64	21.89
139.80	21.96
174.97	21.74
210.13	21.77
262.88	22.12

70°

y^+	u^+
0.22	1.15
0.59	1.14
1.33	1.47
2.08	2.08
2.82	2.43
4.68	4.52
6.54	6.19
17.70	13.01
36.31	20.17
73.52	25.11
110.73	26.08
147.94	26.19
185.15	26.10
222.36	26.13
278.18	26.43

90°

y^+	u^+
0.20	1.07
0.55	1.03
1.25	1.26
1.95	1.79
2.65	2.10
4.40	4.24
6.15	5.89
16.64	13.07
34.13	21.49
69.11	28.49
104.09	30.20
139.06	30.64
174.04	30.63
209.02	30.69
261.49	31.22

100°

y^+	u^+
0.20	1.04
0.54	0.97
1.21	1.17
1.89	1.62
2.57	1.88
4.26	4.03
5.95	5.64
16.11	12.89
33.04	21.96
66.90	29.70
100.76	31.19
134.63	31.75
168.49	31.77
202.35	31.95
253.14	32.51

120°

y^+	u^+
0.32	1.52
0.88	1.75
1.98	2.43
3.09	3.14
4.20	3.49
6.97	5.74
9.75	7.41
26.38	11.01
54.10	13.00
109.54	14.43
164.99	15.43
220.43	16.17
275.88	16.76
331.32	17.15
414.48	17.56

140°

y^+	u^+
0.24	1.29
0.65	1.33
1.47	1.86
2.28	2.48
3.10	2.67
5.15	5.02
7.20	6.60
19.47	11.26
39.94	13.50
80.87	15.10
121.80	16.19
162.74	17.07
203.67	17.71
244.60	18.14
305.99	18.75

110°

y^+	u^+
0.42	1.42
1.15	1.65
2.60	2.28
4.05	2.98
5.50	3.16
9.13	5.11
12.76	6.42
34.52	9.26
70.80	10.63
143.36	11.82
215.92	12.50
288.49	13.24
361.04	13.63
433.61	13.95
542.45	14.35

130°

y^+	u^+
0.29	1.36
0.79	1.57
1.78	2.12
2.77	2.80
3.77	3.14
6.25	5.33
8.73	6.93
23.64	10.80
48.47	13.01
98.15	14.44
147.83	15.51
197.50	16.32
247.18	16.71
296.85	17.28
371.37	17.78

150°

y^+	u^+
0.17	1.17
0.46	1.16
1.04	1.42
1.62	1.86
2.20	2.19
3.65	4.50
5.10	6.25
13.81	11.70
28.32	14.92
57.35	16.92
86.37	18.21
115.39	19.34
144.42	20.00
173.44	20.79
216.98	21.25

s/d=44 Model

160°

y^+	u^+
0.11	1.15
0.30	0.89
0.69	0.80
1.07	0.99
1.45	1.13
2.41	2.61
3.37	4.62
9.12	11.45
18.70	15.36
37.86	17.92
57.03	19.62
76.19	21.12
95.35	22.16
114.52	23.00
143.26	23.96

20°

y^+	u^+
1.00	0.98
2.00	1.93
3.00	2.84
4.00	3.72
5.00	4.56
6.00	5.37
7.00	6.14
8.00	6.87
9.00	7.57
10.00	8.24
11.00	8.87
12.00	9.46
13.00	10.02
14.00	10.55
15.00	11.04

40°

y^+	u^+
1.00	0.99
2.00	1.97
3.00	2.93
4.00	3.87
5.00	4.80
6.00	5.72
7.00	6.62
8.00	7.50
9.00	8.37
10.00	9.22
11.00	10.05
12.00	10.87
13.00	11.68
14.00	12.47
15.00	13.24

170°

y^+	u^+
0.04	1.72
0.12	1.02
0.27	0.46
0.42	0.31
0.56	0.30
0.94	0.66
1.31	1.58
3.54	9.87
7.26	18.17
14.70	23.75
22.15	26.68
29.59	29.69
37.03	32.16
44.47	34.06
55.64	35.87

30°

y^+	u^+
1.00	0.99
2.00	1.96
3.00	2.91
4.00	3.83
5.00	4.74
6.00	5.62
7.00	6.49
8.00	7.33
9.00	8.15
10.00	8.95
11.00	9.73
12.00	10.49
13.00	11.23
14.00	11.94
15.00	12.64

50°

y^+	u^+
1.00	0.99
2.00	1.98
3.00	2.95
4.00	3.91
5.00	4.86
6.00	5.80
7.00	6.72
8.00	7.64
9.00	8.54
10.00	9.43
11.00	10.31
12.00	11.18
13.00	12.04
14.00	12.89
15.00	13.72

60°

y^+	u^+
1.00	1.00
2.00	1.98
3.00	2.96
4.00	3.93
5.00	4.89
6.00	5.85
7.00	6.79
8.00	7.73
9.00	8.66
10.00	9.58
11.00	10.49
12.00	11.39
13.00	12.29
14.00	13.18
15.00	14.05

80°

y^+	u^+
1.00	1.00
2.00	1.99
3.00	2.99
4.00	3.98
5.00	4.97
6.00	5.95
7.00	6.93
8.00	7.91
9.00	8.89
10.00	9.86
11.00	10.83
12.00	11.80
13.00	12.76
14.00	13.73
15.00	14.69

100°

y^+	u^+
1.00	1.00
2.00	2.01
3.00	3.02
4.00	4.03
5.00	5.04
6.00	6.06
7.00	7.08
8.00	8.11
9.00	9.14
10.00	10.17
11.00	11.21
12.00	12.25
13.00	13.29
14.00	14.34
15.00	15.39

70°

y^+	u^+
1.00	1.00
2.00	1.99
3.00	2.98
4.00	3.96
5.00	4.94
6.00	5.91
7.00	6.88
8.00	7.84
9.00	8.79
10.00	9.75
11.00	10.69
12.00	11.63
13.00	12.57
14.00	13.50
15.00	14.43

90°

y^+	u^+
1.00	1.00
2.00	2.00
3.00	3.00
4.00	4.00
5.00	5.00
6.00	6.00
7.00	7.00
8.00	8.00
9.00	9.00
10.00	10.00
11.00	11.00
12.00	12.00
13.00	13.00
14.00	14.00
15.00	15.00

110°		120°		130°	
y^+	u^+	y^+	u^+	y^+	u^+
1.0	1.00	1.0	1.00	1.0	1.00
2.0	2.00	2.0	2.00	2.0	2.00
3.0	2.99	3.0	2.99	3.0	2.99
4.0	3.95	4.0	3.95	4.0	3.95
5.0	4.87	5.0	4.87	5.0	4.86
6.0	5.72	6.0	5.71	6.0	5.70
7.0	6.50	7.0	6.48	7.0	6.45
8.0	7.19	8.0	7.16	8.0	7.12
9.0	7.80	9.0	7.76	9.0	7.72
10.0	8.34	10.0	8.30	10.0	8.24
11.0	8.83	11.0	8.77	11.0	8.71
12.0	9.26	12.0	9.19	12.0	9.12
13.0	9.64	13.0	9.57	13.0	9.49
14.0	9.99	14.0	9.91	14.0	9.83
15.0	10.30	15.0	10.22	15.0	10.13
16.0	10.59	16.0	10.50	16.0	10.40
17.0	10.85	17.0	10.75	17.0	10.65
18.0	11.09	18.0	10.99	18.0	10.88
19.0	11.31	19.0	11.21	19.0	11.10
20.0	11.51	20.0	11.41	20.0	11.29
21.0	11.70	21.0	11.59	21.0	11.48
22.0	11.88	22.0	11.77	22.0	11.65
23.0	12.04	23.0	11.93	23.0	11.81
24.0	12.20	24.0	12.08	24.0	11.96
25.0	12.34	25.0	12.23	25.0	12.10
26.0	12.48	26.0	12.36	26.0	12.23
27.0	12.61	27.0	12.49	27.0	12.36
28.0	12.74	28.0	12.61	28.0	12.48
29.0	12.85	29.0	12.73	29.0	12.59
30.0	12.96	30.0	12.84	30.0	12.70
31.0	13.07	31.0	12.94	31.0	12.80
32.0	13.17	32.0	13.04	32.0	12.90
33.0	13.27	33.0	13.14	33.0	12.99
34.0	13.36	34.0	13.23	34.0	13.09
35.0	13.45	35.0	13.32	35.0	13.17
36.0	13.54	36.0	13.40	36.0	13.26
37.0	13.62	37.0	13.49	37.0	13.34
38.0	13.70	38.0	13.56	38.0	13.42
39.0	13.78	39.0	13.64	39.0	13.49
40.0	13.85	40.0	13.71	40.0	13.56
41.0	13.93	41.0	13.78	41.0	13.63
42.0	13.99	42.0	13.85	42.0	13.70
43.0	14.06	43.0	13.92	43.0	13.77
44.0	14.13	44.0	13.98	44.0	13.83
45.0	14.19	45.0	14.05	45.0	13.89
46.0	14.25	46.0	14.11	46.0	13.95
47.0	14.31	47.0	14.17	47.0	14.01
48.0	14.37	48.0	14.23	48.0	14.07
49.0	14.43	49.0	14.28	49.0	14.12
50.0	14.48	50.0	14.34	50.0	14.18

140°		150°		160°	
y^+	u^+	y^+	u^+	y^+	u^+
1.0	1.00	1.0	1.00	1.0	1.00
2.0	2.00	2.0	2.00	2.0	1.99
3.0	2.98	3.0	2.97	3.0	2.92
4.0	3.94	4.0	3.90	4.0	3.74
5.0	4.84	5.0	4.76	5.0	4.44
6.0	5.66	6.0	5.51	6.0	5.02
7.0	6.39	7.0	6.17	7.0	5.51
8.0	7.04	8.0	6.74	8.0	5.92
9.0	7.61	9.0	7.24	9.0	6.27
10.0	8.12	10.0	7.67	10.0	6.58
11.0	8.56	11.0	8.05	11.0	6.86
12.0	8.96	12.0	8.39	12.0	7.10
13.0	9.31	13.0	8.69	13.0	7.32
14.0	9.63	14.0	8.96	14.0	7.52
15.0	9.91	15.0	9.21	15.0	7.70
16.0	10.18	16.0	9.43	16.0	7.87
17.0	10.41	17.0	9.63	17.0	8.02
18.0	10.63	18.0	9.82	18.0	8.17
19.0	10.84	19.0	10.00	19.0	8.30
20.0	11.02	20.0	10.16	20.0	8.43
21.0	11.20	21.0	10.31	21.0	8.55
22.0	11.36	22.0	10.46	22.0	8.67
23.0	11.52	23.0	10.59	23.0	8.78
24.0	11.66	24.0	10.72	24.0	8.88
25.0	11.79	25.0	10.83	25.0	8.98
26.0	11.92	26.0	10.95	26.0	9.07
27.0	12.04	27.0	11.06	27.0	9.17
28.0	12.16	28.0	11.16	28.0	9.25
29.0	12.27	29.0	11.26	29.0	9.34
30.0	12.37	30.0	11.35	30.0	9.42
31.0	12.47	31.0	11.44	31.0	9.50
32.0	12.57	32.0	11.53	32.0	9.57
33.0	12.66	33.0	11.61	33.0	9.65
34.0	12.75	34.0	11.69	34.0	9.72
35.0	12.83	35.0	11.76	35.0	9.79
36.0	12.91	36.0	11.84	36.0	9.85
37.0	12.99	37.0	11.91	37.0	9.92
38.0	13.06	38.0	11.98	38.0	9.98
39.0	13.14	39.0	12.05	39.0	10.05
40.0	13.21	40.0	12.11	40.0	10.11
41.0	13.28	41.0	12.18	41.0	10.16
42.0	13.34	42.0	12.24	42.0	10.22
43.0	13.41	43.0	12.30	43.0	10.28
44.0	13.47	44.0	12.36	44.0	10.33
45.0	13.53	45.0	12.41	45.0	10.39
46.0	13.59	46.0	12.47	46.0	10.44
47.0	13.65	47.0	12.52	47.0	10.49
48.0	13.70	48.0	12.57	48.0	10.54
49.0	13.76	49.0	12.63	49.0	10.59
50.0	13.81	50.0	12.68	50.0	10.64

170°

y^+	u^+
1.0	1.00
2.0	1.80
3.0	2.42
4.0	2.92
5.0	3.34
6.0	3.69
7.0	4.01
8.0	4.28
9.0	4.53
10.0	4.76
11.0	4.96
12.0	5.16
13.0	5.33
14.0	5.50
15.0	5.65
16.0	5.80
17.0	5.94
18.0	6.07
19.0	6.19
20.0	6.31
21.0	6.42
22.0	6.53
23.0	6.63
24.0	6.73
25.0	6.82
26.0	6.91
27.0	7.00
28.0	7.09
29.0	7.17
30.0	7.25
31.0	7.32
32.0	7.40
33.0	7.47
34.0	7.54
35.0	7.61
36.0	7.68
37.0	7.74
38.0	7.80
39.0	7.86
40.0	7.92
41.0	7.98
42.0	8.04
43.0	8.10
44.0	8.15
45.0	8.20
46.0	8.26
47.0	8.31
48.0	8.36
49.0	8.41
50.0	8.45

I. Data Acquisition Program Listings

Note: All the programs were written in C and executed on a UNIX operating system.

CROSS

This is the main program for cross-wire data acquisition, adapted from the single-wire acquisition program, SINGLE, written by J. Seume (1988). It calls the subroutines ENTER_COND, SET_UP_VMX, and ACQUIREX.

ENTER_COND

This subroutine inputs the test conditions specific to the data set, including piston bore, pipe diameter, stroke, test section length, axial probe location, drive speed, and number of readings per cycle.

SET_UP_VMX

This subroutine sets up the NORLAND digital storage oscilloscope on the IEEE interface for data storage and transfer.

ACQUIREX

This subroutine acquires cross-wire anemometer data with the NORLAND, updates quantities for the calculation of streamwise and radial ensemble-averaged and fluctuating velocities, and velocity correlations, and stores them in a file.

AIR_STATEX

This subroutine is called by ACQUIREX and supplies ambient air conditions to the acquisition program.

II. Data Processing Program Listings

PROCESSX

This is the main program for the processing of cross-wire data into traces and profiles of both the streamwise and radial component of ensemble-averaged and rms-velocity fluctuations, and the Reynolds shear stress.

VEL_REDTURB

This data reduction program converts profiles of ensemble-averaged velocity into wall coordinates, iterating on the wall shear stress and y-offset to fit the data to the Couette flow model for turbulent-like flow, including the effect of pressure gradient.

DUPDYP

This subroutine is called by VEL_REDTURB and calculates the slope in wall coordinates based on the Van Driest mixing length model.

VEL_REDLAM

This data reduction program converts profiles of ensemble-averaged velocity into wall coordinates, iterating on the wall shear stress and y-offset to fit the data to the Couette flow model for laminar-like flow, including the effect of pressure gradient.

CROSS

```
#include <stdio.h>
#define void int

/* definition of external variables */
int  ib3;
int  mrad, lastrad, lastcycle, iuerr, irmserr;
int  ndummy = 50;
int  abort, diagnosis, add_to_set, m_cycle, oldfile, old_file = 0;
int  no_inst;
int  m_out = 0, n_out[20];
double  mstroke, mbore, mdiam, mlength, maxial, speed;
double  t_dry, t_wet, p_atm;
int  nread = 1, nlag = 0, mcycle = 0;
char  filename[30];
char  usage[] = "Usage: single [-a runid or -d or -m mcycle or -n or -o m_out n_
out[1...m_out]]\n";

main(argc,argv)
/* Data acquisition main program, "cross", for cross wire anemometer
measurements.  G. Friedman, 8/90 */
/* The program is an adaptation of single.c, written by J. Seume for use
with single wires */

int  argc;
char  *argv[];
{
#include <string.h>

int  i;

extern int  abort, diagnosis, add_to_set, m_cycle, mcycle;
extern int  no_inst;
extern int  m_out, n_out[];
extern char  filename[];

void  set_up_vmx();
void  enter_cond();
void  acquirex();

        /*      setvbuf (stdout,NULL,_IONBF,1); */
        /* Set default values. */
        diagnosis = 0;
        add_to_set = 0;
        m_cycle = 0;
        no_inst = 0;

        /* Read command line for control parameters. */
        while(***++argv == '-')
                {--argc;
                switch(*++*argv)
                {
                case 'a':
                        /* Add to an already existing data set. */
                        add_to_set = 1;
                        strcpy(filename, "/usr/geoff/shwdata/");
                        strcpy(&filename[19], *++argv);
                        --argc;
                         break;
                 case 'd':
                        /* Print out diagnostic information. */
                        diagnosis = 1;
                        break;
                 case 'm':
                        /* Set maximum number of cycles to be acquired
                                to a value different from the default
                                 specified in acquire.c. */
                        m_cycle = 1;
                        mcycle = atoi(*++argv);
                        --argc;
```

```
                break;
        case 'n':
                /* Do not access the A/D converter. */
                no_inst = 1;
                --argc;
                break;
        case 'o':
                /* Set maximum number of cycles to be acquired
                        to a value different from the default
                         specified in acquire.c. */
                m_out = atoi(*++argv);
                if(m_out > 9) {
                        printf("m_out = %d > 9 => Choose ", m_out);
                        printf("m_out <= 9 !\n");
                        }

                for(i = 1; i <= m_out; i++) {
                        n_out[i] = atoi(*++argv);
                        if(diagnosis) printf("n_out[%d] = %d\n",
                                                i, n_out[i]);
                        }
                --argc;
                break;
        default:
                printf("%s", usage);
                break;
        }
        }

/* ******************************** */
/* Calls to the various subroutines */

/* Enter experimental conditions. */
enter_cond();
if(abort) goto the_end;

/* Set up A/D converter. */
if(! no_inst) set_up_vmx();
if(abort) goto the_end;

/* Acquire data and store them in file under
        directory /usr/geoff/shwdata */
acquirex();
if(abort) goto the_end;
/* Address for abort sequence. */
the_end:;
}
```

ENTER_COND

```
enter_cond()
/* Enter nominal test conditions. */
{
#include <stdio.h>
#include <math.h>
#include <string.h>

#define NSTROKE 6
#define NBORE 4
#define NDIAM 3
#define NLENGTH 9
#define NAXIAL 21
#define IDLENGTH 10
#define NDUMMY 50
#define MCYCLE 100
#define PI (4. * atan(1.0))
#define NU 16.e-06 /* nominal kinematic viscosity */

extern int  abort, add_to_set; /* = 1 =>  Add runs to an existing
                                          set of data. */

extern int  diagnosis;
extern int  old_file;
extern int  m_cycle, mcycle;
extern int  oldfile; /* = 1 => File existed and will be updated. */
extern double  mstroke, mbore, mdiam, mlength, maxial, speed;
extern int  nread; /* number of readings per cycle */
extern int  nlag; /* number of sample pulses (readings) by which TDC trigger
                     lags TDC */
extern char  filename[];

FILE  *storefile;

int  i, istroke, ibore, idiam, ilength, iaxial, ispeed, inerr;
double  Remax, Va, Ar, loverd, xoverl, xoverd, uavemax;
static double  stroke[NSTROKE] = {0.0,14.0,9.9,7.0,4.95,3.5};
static double  lag_angle[NSTROKE] = {0.0,0.0,180.,90.,270.,210.};
       /* lag_angle = angle by which TDC trigger signal lags TDC */
static double  bore[NBORE] = {0.0,14.0,8.5,5.0};
       /* Ideal diameters would be: {0.0,14.142,8.409,5.0}; */
static double  diam[NDIAM] = {0.0,1.5,2.125};
static int  nlengths[NDIAM][NLENGTH] =
                        {{0},
                         {0,1,2,3,4,5,6},
                         {0,7,8}
                        };
static double  length[NLENGTH] = {0.0,42.,60.,90.,102.72,120.,127.5,59.5,127.5}
static double  axial[NLENGTH][NAXIAL] =
                        {{0.0},
                         {0,1.5,3,6,12,24,30.72},
                         {0,1.5,3,6,12,24,30,36,48,51.36,54,57,58.5},
                         {0,1.5,3,6,12,24,30,36,45,48,51.36,54,57,58.5,
                                66,72,78,84,87,88.5},
                         {0,1.5,3,6,12,24,30,36,48,51.36,54,57,58.5,
                                72,78.72,90.72,96.72,99.72,101.22},
                         {0,1.5,3,6,12,24,30,36,48,51.36,54,57,58.5,
                                72,96,108,114,117,118.5},
                         {0,1.5,3,6,12,24,30,36,48,51.36,54,57,58.5,
                                63.75,72,79.5,103.5,115.5,121.5,124.5,126},
                         {0,2.125,4.25,8.5,17,25.5,29.75,34,
                                42.5,51,55.25,57.375},
                         {0,2.125,4.25,8.5,17,25.5,29.75,34,
                                42.5,51,55.25,57.375,
                                63.75,93.5,110.5,119.0,123.25,125.375}
                        };
static char  dummy[NDUMMY]; /* dummy array to keep room for further
                               descriptors for runs */

char  runid[10], resp[5];
char  year[3], month[3];
```

```
if(! add_to_set) {

        /* Set the maximum number of cycles to be acquired. */
        if(!m_cycle) mcycle = MCYCLE;

        /* Read run identification and check whether the corresponding
              file already exists. */
        for(inerr = 1; inerr; ) {
                printf("\nEnter run identification: \n\n");
                printf("(Use the format mmddyyss where: \n");
                printf(" mm = month, dd = day, yy = year,\n");
                printf(" and ss = a sequence number of the day's");
                printf(" runs)\n");
                scanf("%s", runid);
                printf("mm = %c%c    ", runid[0], runid[1]);
                printf("dd = %c%c    ", runid[2], runid[3]);
                printf("yy = %c%c    ", runid[4], runid[5]);
                printf("ss = %c%c\n ", runid[6], runid[7]);
                strcpy(filename,"/usr/geoff/shwdata/");
                strcpy(&filename[19],runid);
                printf(" The filename is '%s'\n\n", filename);
                storefile = fopen(filename, "r+");
                if(storefile == NULL)
                        {printf(" This file did not exist but may now ");
                        printf("be created.\n");
                        oldfile = 0;}
                else
                        {printf(" This file exists and data may be added.\n")
                        oldfile = 1;}
                printf("\n Entry correct? (y or n)\n");
                scanf("%s", resp);
                if(resp[0] == 'y' || resp[0] == 'Y')
                        {inerr = 0;}
                else if(resp[0] == 'n' || resp[0] == 'N')
                        {inerr = 1;}
                else
                        {printf(" Respond with y, Y for 'yes' ");
                        printf("or with n, N for 'no' next time.\n");}
                }
         if(! oldfile)
                {printf(" File has now been created.\n");
                storefile = fopen(filename,"a+");}
         fclose(storefile);
         /* Read nominal test conditions from key board. */
         if(! oldfile) {
                printf("\nEnter nominal test conditions.\n");
                for(inerr = 1; inerr; )
                        {printf("\nStroke: \n");
                        printf("        stroke          code\n");
                        printf("        length          number\n");
                        printf("        (inches)\n");
                        for(i = 1; i <= NSTROKE-1; i = i + 1)
                                printf("\t%7.3f\t\t%d\n", stroke[i], i);
                        printf(" Enter code for stroke length.\n");
                        printf(" [If the desired stroke is not listed, ");
                        printf("enter a '0'.]\n");
                        while(scanf("%d", &istroke) == 0){
                                getchar();
                                printf(" Enter an integer code number!\n");
                                }
                        if(istroke == 0){
                                printf(" Enter stroke value in inches.\n");
                                scanf("%lf", &stroke[0]);
                                }
                        printf(" Stroke length = %7.3f", stroke[istroke]);
                        printf(" Entry correct? (y or n)\n");
                        scanf("%s", resp);
                        if(resp[0] == 'y' || resp[0] == 'Y')
                                inerr = 0;
                        else if(resp[0] == 'n' || resp[0] == 'N')
                                inerr = 1;
                        else {
                                printf(" Respond with y, Y for 'yes' ");
```

```
                printf("or with n, N for 'no' next time.\n");
                }
        }
for(inerr = 1; inerr; )
        {printf("\nBore: \n");
        printf("        bore            code\n");
        printf("        (inches)        number\n");
        for(i = 1; i <= NBORE-1; i = i + 1)
                printf("\t%7.3f \t%d\n", bore[i], i);
        printf(" Enter code for bore.\n");
        printf(" [If the desired bore is not listed, ");
        printf("enter a '0'.]\n");
        while(scanf("%d", &ibore) == 0)
                {getchar();
                printf(" Enter an integer code number!\n");}
        if(ibore == 0)
                {printf(" Enter bore value in inches.\n");
                scanf("%lf", &bore[0]);}
        printf(" Bore = %7.3f in\n", bore[ibore]);
        printf(" Entry correct? (y or n)\n");
        scanf("%s", resp);
        if(resp[0] == 'y' || resp[0] == 'Y')
                {inerr = 0;}
        else if(resp[0] == 'n' || resp[0] == 'N')
                {inerr = 1;}
        else
                {printf(" Respond with y, Y for 'yes' ");
                printf("or with n, N for 'no' next time.\n");}
        }
for(inerr = 1; inerr; )
        {printf("\nTube diameter: \n");
        printf("        diameter        code\n");
        printf("        (inches)        number\n");
        for(i = 1; i <= NDIAM-1; i = i + 1)
                printf("\t%7.3f \t%d\n", diam[i], i);
        printf(" Enter code for diameter.\n");
        printf(" [If the desired diameter is not listed, ");
        printf("enter a '0'.]\n");
        while(scanf("%d", &idiam) == 0)
                {getchar();
                printf(" Enter an integer code number!\n");}
        if(idiam == 0)
                {printf(" Enter diameter value in inches.\n");
                scanf("%lf", &diam[0]);}
        printf(" Tube diameter = %7.3f in\n", diam[idiam]);
        printf(" Entry correct? (y or n)\n");
        scanf("%s", resp);
        if(resp[0] == 'y' || resp[0] == 'Y')
                {inerr = 0;}
        else if(resp[0] == 'n' || resp[0] == 'N')
                {inerr = 1;}
        else
                {printf(" Respond with y, Y for 'yes' ");
                printf("or with n, N for 'no' next time.\n");}
        }
for(inerr = 1; inerr;)
        {printf("\nTest section length: \n");
        printf("\ttube\tcode\tlength\n");
        printf("\tlength\tnumber\tover\n");
        printf("\t(inches)\tdiameter\n");
        for(i = 1; i <= NLENGTH-1; i = i + 1)
                printf("\t%7.2f\t   %d\t%7.2f\n",
                        length[nlengths[idiam][i]],
                        nlengths[idiam][i],
                        length[nlengths[idiam][i]] /
                                diam[idiam]);
        printf(" Enter code for test section length.\n");
        printf(" [If the desired length is not listed, ");
        printf("enter a '0'.]\n");
        while(scanf("%d", &ilength) == 0)
                {getchar();
                printf(" Enter an integer code number!\n");}
        if(ilength == 0)
```

```
                {printf(" Enter tube length in inches.\n");
                scanf("%lf", &length[0]);}
        printf(" Test section length = %7.2f in\n",
                                        length[ilength]);
        printf(" Entry correct? (y or n)\n");
        scanf("%s", resp);
        if(resp[0] == 'y' || resp[0] == 'Y')
                {inerr = 0;}
        else if(resp[0] == 'n' || resp[0] == 'N')
                {inerr = 1;}
        else
                {printf(" Respond with y, Y for 'yes' ");
                printf("or with n, N for 'no' next time.\n");}
        }
for(inerr = 1; inerr; )
        {printf("\nAxial probe location: \n");
        printf("        axial           code\n");
        printf("        distance        number\n");
        printf("        (inches)\n");
        for(i = 1; i <= NAXIAL-1; i = i + 1)
                printf("       %6.3f            %d\n",
                                        axial[ilength][i], i);
        printf(" Enter code for axial distance.\n");
        printf(" [If the desired distance is not listed, ");
        printf("enter a '0'.]\n");
        while(scanf("%d", &iaxial) == 0)
                {getchar();
                printf(" Enter an integer code number!\n");}
        if(iaxial == 0)
                {printf(" Enter axial distance in inches.\n");
                scanf("%lf", &axial[ilength][0]);}
        printf(" Probe location = %6.3f in\n",
                        axial[ilength][iaxial]);
        printf(" Entry correct? (y or n)\n");
        scanf("%s", resp);
        if(resp[0] == 'y' || resp[0] == 'Y')
                {inerr = 0;}
        else if(resp[0] == 'n' || resp[0] == 'N')
                {inerr = 1;}
        else
                {printf(" Respond with y, Y for 'yes' ");
                printf("or with n, N for 'no' next time.\n");}
        }

for(inerr = 1; inerr; )
        {printf("\nDrive speed:\n");
        printf("Code to enter drive shaft rpm = '1'\n");
        printf("Code to enter flywheel frequency in Hz =");
        printf(" '2'\n");
        printf(" Enter code for entry.\n");
        while(scanf("%d", &ispeed) == 0)
                {getchar();
                printf(" Enter an integer code number!\n");}
        if(ispeed == 1)
                {printf(" Enter shaft speed in rpm.\n");
                scanf("%lf", &speed);
                speed = speed / 240.;}
        else if (ispeed == 2)
                {printf(" Enter flywheel frquency in Hz.\n");
                scanf("%lf", &speed);}
        printf(" Shaft speed = %6.1f rpm    ", (speed * 240.));
        printf(" Flywheel frequency = %7.3f Hz\n", speed);
        printf(" Entry correct? (y or n)\n");
        scanf("%s", resp);
        if(resp[0] == 'y' || resp[0] == 'Y')
                {inerr = 0;}
        else if(resp[0] == 'n' || resp[0] == 'N')
                {inerr = 1;}
        else
                {printf(" Respond with y, Y for 'yes' ");
                printf("or with n, N for 'no' next time.\n");}
        }
```

```
/* The following lines is used to incorporate shaft-angle
   encoder signals with less than 720 pulses per revolution. */
for(inerr = 1; inerr; ) {
        printf("\nEnter number of readings per cycle:\n");
        while(scanf("%d", &nread) == 0)
                {getchar();
                printf(" Enter an integer number!\n");}
        printf(" %d readings per cycle will be taken.\n",
                                                        nread);
        printf(" Divide 720 pulses by %f.\n",
                                        720./(double)nread);
        if(speed * (double) nread > 25.)
                printf(" Frequency of readings is %f Hz!\n",
                                        speed * (double) nread);
        nlag = (int)((double)nread * lag_angle[istroke] / 360.);
        if(diagnosis) printf("lag angle = %lf, nlag = %d\n",
                                lag_angle[istroke], nlag);
        printf(" Entry correct? (y or n)\n");
        scanf("%s", resp);
        if(resp[0] == 'y' || resp[0] == 'Y')
                {inerr = 0;}
        else if(resp[0] == 'n' || resp[0] == 'N')
                {inerr = 1;}
        else
                {printf(" Respond with y, Y for 'yes' ");
                printf("or with n, N for 'no' next time.\n");}
        }

/* Alternative: Dummy entry of shaft-angle encoder pulses. */
/* NOT USED:
nread = 720;
nlag = 2 * (int)(lag_angle[istroke]);
if(diagnosis) printf("lag angle = %lf, nlag = %d\n",
                                lag_angle[istroke], nlag);
*/

for(inerr = 1; inerr; ) {
        printf("\nEnter a 50 character comment line:\n");
        printf("(Use _ instead of blank spaces!)\n");
        for(i = 1; i <= 5; i++) printf("1234567890");
        printf("\n");
        for(i = 1; i <= 6; i++) printf("%d         ", i-1);
        printf("\n");
        scanf("%s", dummy);
        printf("Comment line: \n%s", dummy);
        printf("\n Entry correct? (y or n)\n");
        scanf("%s", resp);
        if(resp[0] == 'y' || resp[0] == 'Y')
                {inerr = 0;}
        else if(resp[0] == 'n' || resp[0] == 'N')
                {inerr = 1;}
        else
                {printf(" Respond with y, Y for 'yes' ");
                printf("or with n, N for 'no' next time.\n");}
        }

/* Convert to SI base units. */
mstroke = stroke[istroke] * 0.0254;
mbore = bore[ibore] * 0.0254;
mdiam   = diam[idiam] * 0.0254;
mlength = length[ilength] * 0.0254;
maxial = axial[ilength][iaxial] * 0.0254;

/* Write runid and parameters to new data file. */
storefile = fopen(filename, "r+");
fwrite(runid,sizeof(char),10,storefile);
fwrite(&mstroke,sizeof(double),1,storefile);
fwrite(&mbore,sizeof(double),1,storefile);
fwrite(&mdiam,sizeof(double),1,storefile);
fwrite(&mlength,sizeof(double),1,storefile);
fwrite(&maxial,sizeof(double),1,storefile);
fwrite(&speed,sizeof(double),1,storefile);
fwrite(&nread,sizeof(int),1,storefile);
```

```
        if(! old_file) {
                fwrite(&mcycle,sizeof(int),1,storefile);
                fwrite(dummy,sizeof(char),NDUMMY,storefile);
                }
        fclose(storefile);
        }
else {
        /* Read parameters from old data file and provide a summary in
                English units. */
        storefile = fopen(filename, "r+");
        fread(runid,sizeof(char),10,storefile);
        fread(&mstroke,sizeof(double),1,storefile);
        fread(&mbore,sizeof(double),1,storefile);
        fread(&mdiam,sizeof(double),1,storefile);
        fread(&mlength,sizeof(double),1,storefile);
        fread(&maxial,sizeof(double),1,storefile);
        fread(&speed,sizeof(double),1,storefile);
        fread(&nread,sizeof(int),1,storefile);
        strncpy(year,&runid[4],2);
        strncpy(month,&runid[0],2);
        if(atoi(year) == 88 && atoi(month) < 6) old_file = 1;
        if(! old_file) {
                fread(&mcycle,sizeof(int),1,storefile);
                fread(dummy,sizeof(char),NDUMMY,storefile);
                }
        fclose(storefile);

        /* Determine the lag in terms of number of pulses. */
        for(i = 0; i <= NSTROKE; i++) {
                if(stroke[i] > (0.95 * mstroke / 0.0254)
                   && stroke[i] < (1.05 * mstroke / 0.0254)) {
                        nlag = (int)((double)nread * lag_angle[i]
                                                        / 360.);
                        istroke = i;
                        }
                }
        if(diagnosis) printf("lag angle = %lf, nlag = %d\n",
                                        lag_angle[istroke], nlag);

        printf("\n Input data summary in English units:\n");
        printf(" stroke =       %6.1f in\n", mstroke/0.0254);
        printf(" bore   =       %6.1f in\n", mbore/0.0254);
        printf(" test section diameter =\t%6.1f in\n", mdiam/0.0254);
        printf(" test section length   =\t%6.1f in\n", mlength/0.0254)
        printf(" axial location        =\t%6.1f in\n", maxial/0.0254);
        printf(" drive shaft speed =    %6.1f rpm\n", speed*240.);
        printf(" %d readings per cycle\n", nread);
        printf(" [Divide 720 by %f.]\n", 720./(double)nread);
        printf(" %s\n", dummy);
        printf("Type 'c <CR>' to continue.\n");
        scanf("%s", resp);
        }

/* Echo print input data in SI units. */
printf("\n Input data summary in SI units:\n");
printf(" stroke =       %6.1f mm\n", mstroke*1000);
printf(" bore   =       %6.1f mm\n", mbore*1000);
printf(" test section diameter =        %6.1f mm\n", mdiam*1000);
printf(" test section length   =        %6.1f mm\n", mlength*1000);
printf(" axial location        =        %6.1f mm\n", maxial*1000);
printf(" frequency =    %6.3f Hz\n", speed);
/* estimate of the amplitude of the bulk-mean velocity */
printf("Estimated amplitude of the bulk-mean velocity = ");
uavemax = PI * speed * mstroke * (mbore*mbore) / (mdiam*mdiam);
printf("%5.2f m/sec\n", uavemax);

/* Calculate and print similarity parameters. */
printf("\n Nominal similarity parameters:\n");
printf(" Remax =        %10.2e\n",
                Remax = PI*mbore*mbore*speed*mstroke / mdiam / NU);
printf(" Va =           %6.1f\n", Va = 0.5*PI*speed*mdiam*mdiam / NU);
printf(" Ar =           %6.2f\n",
                Ar = mbore*mbore/(mdiam*mdiam)*mstroke/mlength);
```

```
        printf(" l/d =          %6.1f\n", loverd = mlength / mdiam);
        printf(" x/l =          %6.3f\n", xoverl = maxial / mlength);
        printf(" x/d =          %6.1f\n", xoverd = maxial / mdiam);
        }
else {
        storefile = fopen(filename, "r+");
        if(storefile == NULL)
                {printf(" This file does not exist.\n");
                abort = 1;}
        else {
                oldfile = 1;

                /* Read parameters from old data file. */
                fread(runid,sizeof(char),10,storefile);
                fread(&mstroke,sizeof(double),1,storefile);
                fread(&mbore,sizeof(double),1,storefile);
                fread(&mdiam,sizeof(double),1,storefile);
                fread(&mlength,sizeof(double),1,storefile);
                fread(&maxial,sizeof(double),1,storefile);
                fread(&speed,sizeof(double),1,storefile);
                fread(&nread,sizeof(int),1,storefile);
                strncpy(year,&runid[4],2);
                strncpy(month,&runid[0],2);

                /* Determine the lag in terms of number of pulses. */
                for(i = 0; i <= NSTROKE; i++) {
                        if(stroke[i] > (0.95 * mstroke / 0.0254)
                           && stroke[i] < (1.05 * mstroke / 0.0254)) {
                                nlag = (int)((double)nread * lag_angle[i]
                                                                / 360.);
                                istroke = i;
                                }
                        }
                if(diagnosis) printf("lag angle = %lf, nlag = %d\n",
                                                lag_angle[istroke], nlag);

                /* Check whether this is a file of the old format. */
                if(atoi(year) == 88 && atoi(month) < 6) old_file = 1;
                if(! old_file) fread(dummy,sizeof(char),NDUMMY,storefile);

                fclose(storefile);
                }
        }
}
```

SET_UP_VMX

```
set_up_vmx()
/* Set up NORLAND Prowler on IEEE interface. */
/* This program is an adaptation of set_up_vm for cross wires */
{
#include <gpib.h>
#include <stdio.h>
#include <string.h>

extern int  ib3;
char  set3[100];
char  resp[100];

/* Send message to screen. */
printf("NORLAND Prowler will now be configured.\n");

/* Identify device and set up interface. */
ib3 = ibfind("/dev/ib3"); /* Define device ID. */
ibtmo(ib3,14); /* Timeout = 30sec */

/* Set controls on device and check interface communications. */

/* Generate string of control commands to be sent to device. */
strcpy(set3,"]LA"); /* Beeper off */
strcat(set3,"Y"); /* ACQ. MODE */
strcat(set3,"E"); /* TRIGGERED HOLD */
strcat(set3,"M4096="); /* BLOCK SIZE = 4096 */
strcat(set3,"L0="); /* SAMPLE INTERVAL = EXT. */
strcat(set3,"Z"); /* TRIGGER SETUP */
strcat(set3,"G4096="); /* EXTERNAL TRIGGER DELAY = 4096 */
strcat(set3,"ME"); /* SOURCE = EXTERNAL*/
strcat(set3,"["); /* A SETUP */
strcat(set3,"@"); /* ACTIVE */
strcat(set3,"C2="); /* RANGE = 2 */
strcat(set3,"E0="); /* BIAS = 0% */
strcat(set3,"GC"); /* COUPLING = DC */
strcat(set3,"\\"); /* B SETUP */
strcat(set3,"@"); /* ACTIVE */
strcat(set3,"C2="); /* RANGE = 2 */
strcat(set3,"E0="); /* BIAS = 0% */
strcat(set3,"GC"); /* COUPLING = DC */

ibwrt(ib3,set3,strlen(set3)); /* Send string to device. */
while(ERR & ibsta){
        switch (iberr)
        {
        case 0:
                {printf("iberr = %d:  operating system error\n", iberr);
                printf(" UNIX error code = %d\n", ibcnt);
                break;}
        case 1:
                {printf("iberr = %d:  GPIB must be in charge.\n", iberr);
                break;}
        case 2:
                {printf("iberr = %d:  Write function detected ",
                                                        iberr);
                printf("no listeners.\n");
                break;}
        case 3:
                {printf("iberr = %d:  interface board", iberr);
                printf(" not addressed correctly.\n");
                break;}
        case 4:
                {printf("iberr = %d:  invalid arg. to fctn call\n", iberr);
                break;}
        case 5:
                {printf("iberr = %d:  GPIB-board must be ", iberr);
                printf("System Active Controller.\n");
                break;}
        case 6:
```

```
                {printf("iberr = %d:  I/O operation aborted.\n", iberr);
                break;}
        case 7:
                {printf("iberr = %d:  Interface board does not ", iberr);
                printf("exist.\n");
                break;}
        case 10:
                {printf("iberr = %d:  I/O started before ", iberr);
                printf("previous operation completed. \n");
                break;}
        case 11:
                {printf("iberr = %d:  no capability for ", iberr);
                printf("operation\n");
                break;}
        case 14:
                {printf("iberr = %d:  command error during ", iberr);
                printf("device call\n");
                break;}
        case 15:
                {printf("iberr = %d:  Serial Poll status ", iberr);
                printf("byte lost\n");
                break;}
        case 16:
                {printf("iberr = %d:  SQR remains asserted.\n", iberr);
                break;}
        }
        printf(" => Check NORLAND and connections.\n");
        printf("    Type 'c <CR>' when you are ready.\n");
        scanf("%s",resp);
        ibwrt(ib3,set3,strlen(set3)); /* Send string to device. */
        }
}
```

ACQUIREX

```
acquirex()
/* This program acquires cross-wire anemometer data with a NORLAND Prowler
        digital storage oscilloscope, updates quantities for the calculation of
        mean and fluctuating velocities, and correlations, and stores them in
        a file.  G. Friedman, 8/90 */
/* This program is based on a version of acq_um.c, with wires A and B, written
        by J. Seume, 1988 */
{
#include <gpib.h>
#include <stdio.h>
#include <string.h>
#include <math.h>

#define  MREAD 181          /* maximum number of data array entries */
#define  MREAD2 721         /* shaft-angle encoder angles in NORLAND arrays */
#define  MREAD3 903         /* ueffA storage for 180 pts x 5 cycles */
#define  AMAX 4096          /* number of entries in NORLAND data array */
#define  SMAX 8452          /* number of entries in string received from NORLAND */
#define  MCYCLE 50          /* maximum number of cycles */
#define  MRAD 30            /* maximum number of radial probe locations */
#define  STRLNG 30
#define  NDUMMY 50          /* This parameter is also needed in air_statex. */
#define  PI 3.14159265

extern int  ib3;
extern int  diagnosis, m_cycle, old_file, no_inst;
extern int  m_out, n_out[];
extern double  mstroke, mbore, mdiam, mlength, maxial, speed;
extern double  t_dry, t_wet, p_atm;  /* air conditions during run */
extern int  nread, mcycle;
extern int  nlag;           /* number of sample pulses (readings) by which the
                               TDC trigger lags TDC */
extern char  filename[];

FILE  *storefile, *densfile;
FILE  *umfile, *vmfile;
int     inerr,              /* error in input data */
        irad,               /* number of current radial probe location */
        iuerr = 0, irmserr = 0, /*sequence number of angle at which
                                  maximum error in umean and urms
                                  occurred in this cycle */
        next_round = 1,     /* acquire another round of data */
        round,              /* A "round" of data are those data points in the
                               NORLAND's buffer that represent a complete cycle. */
        cy_in_buf,          /* number of cycles in the NORLAND's buffer */
        morerad = 1,        /* acquire at more radial locations */
        fin_prt = 0,        /* = 1  =>  final printout, lastcy >= mcycle */
        int_prt = 0,        /* = 1  =>  intermediate printout, m_out >= 1 */
        iout = 1,           /* index for n_out[] */
        icycle,             /* number of current cycle */
        lastcy,             /* number acquired at one radial location */
        iangle,             /* number of current crank location */
        i,kcount,jcount,icount,count,ucount,count3,     /* auxiliary counters *
        iter
        ;
double  dist, distance, relrad,         /* probe location */
        accuracy = 0.005,               /* accuracy of umean */

        acalib,bcalib,ncalib,   /* general calibration constants for
                                   substitution of values below */
        ninv,                   /* inverse of ncalib */

/*      ********************************************************        */
        /*              xwire calibrations                              */
        /* calibration of 7/10/90, cross-wire #35333 for ujet <= 25 m/sec */
        acalibA = 2.94934, bcalibA = 1.85552, ncalibA = 0.435,
        acalibB = 3.18061, bcalibB = 1.96477, ncalibB = 0.435,
```

```
        /* calibration of 6/20/90, cross-wire #35333 for ujet <= 17 m/sec
        acalibA = 2.83801, bcalibA = 1.82569, ncalibA = 0.435,
        acalibB = 3.15662, bcalibB = 1.91373, ncalibB = 0.435, */

        /* calibration of cross-wire #44135 for 0.0 m/s to 17.5 m/s (3/14/90)
        /* Wire A is the same as wire 1 */
        /* acalibA = 4.08779, bcalibA = 1.35912, ncalibA = .5,
        acalibB = 4.24447, bcalibB = 1.41936, ncalibB = .5, */
                                /* calibration constants:
                                E^2 = acalib + bcalib * (veleff)^ncalib */
/*      ************************************************************      */

        ref_temp = 27.2,        /* dry-bulb temperature at calibration
                                in degrees Celsius */
        ref_press = 0.991e+05,  /* atmospheric pressure at calibration in Pa *
        ref_dens,               /* density at calibration */
        voffset,                /* voltage that was subtracted from
                                hot wire signal during conditioning */
        voffsetA, voffsetB,
        vgain,                  /* multiplication factor that was
                                applied to voltage during signal conditioning
        vgainA = 1.0, vgainB = 1.0,
        voltage,                /* hot-wire bridge output voltage */
        trigang = 0.0,          /* angle after TDC at which the TDC
                                trigger is actusted  (in degrees) */
        base, argument,         /* auxiliary variables */
        veleff,                 /* instantaneous effective velocity in m/s */
        cycles,                 /* number of cycles processed at this point */
        maxuerr = 0.0,          /* accuracy of umean */
        mxrmserr = 0.0,
        um,                     /* auxiliary variable for um and vm */
        vm,
        u,v,                    /* auxiliary variables for u and v,
                                based upon ueffA and ueffB */
        theta = 45.,            /* between sensor and the normal to flow */
        angle,                  /* theta, in radians */
        kt = 0.135,             /* coefficient for the tangential cooling,
                                based on Champagne's work, for l/d = 330 */

        ktAterm,ktBterm,        /* terms from the iteration for u and v */
        sint,cost,
        ueffAnew, ueffBnew,
        vnew, unew,
        epsilonA, epsilonB,
        rhsA, lhsA, rhsB, lhsB,

        off_set,                /* auxiliary voltage offset variable */
        volt_corr,              /* voltage correction to account for
                                ambient temperature difference between
                                calibration and run */
        t_sensor = 250.,        /* sensor temperature in degrees Celsius */
        gas_const = 8315.,      /* universal gas constant in kg/kmol/K */
        air_mwt = 28.96,        /* molecular weight in kg/kmol for air
                                with mole fractions of:
                                        N2 - 0.7809
                                        O2 - 0.2095
                                        Ar - 0.0096     */
        density,                /* current air density */
        mean_dens;              /* density at atmospheric pressure */

static double   sumu[MREAD],    /* sum of instantaneous velocities, for u
                                also: values of umean[], re-sorted for
                                lag in crank-angle */
                sumv[MREAD],
                sumu2[MREAD],   /* sum of squares of instantaneous velocities,
                                also: values of urms[], re-sorted for
                                lag in crank-angle */
                sumv2[MREAD],
                sumuv[MREAD],   /* sum of product of instantaneous velocities *
                old_su[MREAD], old_su2[MREAD],          /* current values
                                                        of sumu, sumu2 */
                old_sv[MREAD],old_sv2[MREAD], old_suv[MREAD],
                ueffA[MREAD3],ueffB,    /* effective velocities from
                                        wires A and B, from which get u, v */
```

```
                umean[MREAD],           /* ensemble averaged velocity, also
                                           used to store the density correction
                                           factor */
                vmean[MREAD],
                urms[MREAD],            /* velocity fluctuation */
                vrms[MREAD],
                upvp[MREAD]             /* Reynolds' shear stress */
                ;
char  resp[5];
char  status_byte;
char  set3[STRLNG], acquire[2], beeper_on[4];
static char  set_range[100], rdcmdA[STRLNG],rdcmdB[STRLNG];

int  *iptr, ifactor1, hexdigit, sign, ivalue;
int  first_time;
int  bit7, bit6, bit5, bit4, bit3, bit2, bit1, bit0;
int  air_statex();
double  *dptr, factor1, factor2, factor, offset;
static char  readstrA[SMAX],readstrB[SMAX], digit[2];
static char  *ptrdstr;                  /* pointer to readstr */
static char  dummy[NDUMMY], buffer[NDUMMY];
/*              Prepare for communication with the NORLAND              *

/* Generate strings for ACQUIRE and Beeper Off commands. */
strcpy(acquire,"R");
strcpy(beeper_on,"]LC");                /* Beeper on */

/* Generate command string for data transfer from NORLAND of array A. */
strcpy(rdcmdA,"_");                     /* I/O */
strcat(rdcmdA,"K");                     /* TRANSFER */
strcat(rdcmdA,"C");                     /* OUTPUT */
strcat(rdcmdA,"G");                     /* XFAST BINARY */
strcat(rdcmdA,"A");                     /* ARRAY A */

/* Generate command string for data transfer from NORLAND of array B. */
strcpy(rdcmdB,"_");                     /* I/O */
strcat(rdcmdB,"K");                     /* TRANSFER */
strcat(rdcmdB,"C");                     /* OUTPUT */
strcat(rdcmdB,"G");                     /* XFAST BINARY */
strcat(rdcmdB,"C");                     /* ARRAY B */

/* Generate and send command string to reset the voltage ranges. */
strcpy(set_range,"[C");                 /* A SETUP RANGE */
strcat(set_range,"5=");                 /* 5 V */
strcat(set_range,"\\C");                /* B SETUP RANGE */
strcat(set_range,"5=");                 /* 5 V */
ibwrt(ib3,set_range,strlen(set_range));
/*              Input and print run information prior to
                entering the main loop for ensemble averaging           *

/* Calculate number of cycles completely represented in one NORLAND buffe
based upon 720 samples/cycle via the shaft angle encoder */
cy_in_buf = 4096 / (MREAD2-1);

/* Print out nlag. */
if(diagnosis) printf("nlag = %d\n", nlag);

/* Print out cycles at which intermediate results are to be stored. */
if(diagnosis) {
        for(i = 1; i <= m_out; i++) {
                printf("n_out[%d] = %d\n", i, n_out[i]);
                }
        }

/* Set the maximum number of cycles to be acquired. */
if(!m_cycle) mcycle = MCYCLE;

/* Convert temperatures from Celsius to Kelvin. */
ref_temp += 273.15;
t_sensor += 273.15;

/* Enter voltage offsets for wires A and B. */
for(count = 1; count<=2; count++){
```

```
  for(inerr = 1; inerr; ){
        if(count == 1) printf("\nVoltage offset for wire A:\n");
                       printf("\nVoltage offset for wire B:\n");
        printf(" Enter absolute voltage value.\n");
        if(count == 1) scanf("%lf", &voffsetA);
        else
                       scanf("%lf", &voffsetB);
        if(count == 1) printf(" Voltage offset = %7.4g V\n", voffsetA);
        else
                       printf(" Voltage offset = %7.4g V\n", voffsetB);
        printf(" Entry correct? (y or n)\n");
        scanf("%s", resp);
        if(resp[0] == 'y' || resp[0] == 'Y')
                {inerr = 0;}
        else if(resp[0] == 'n' || resp[0] == 'N')
                {inerr = 1;}
        else
                {printf(" Respond with y, Y for 'yes' ");
                printf("or with n, N for 'no' next time.\n");}
          }
        }

        /*  The program assumes that theta = 45 degrees for the
        xwire probe.  The iteration section must be changed if this
        is not the case. */
        printf("\nNote that the program assumes theta = 45.\n");

/*      ************************************************************        */
/*                       Loop for radial postions                           */

for( ; 1; ) {

        /* Initialize arrays for summation and velocity calculation. */
        for(i = 0; i <= MREAD; i++){
                sumu[i] = 0.0;
                sumv[i] = 0.0;
                sumu2[i] = 0.0;
                sumv2[i] = 0.0;
                sumuv[i] = 0.0;
                }

        /* Wake up the operator with a bell. */
        for(i = 0; i < 10; i++) printf("%c", '\007');

        /* Enter a comment line. */
        for(inerr = 1; inerr; ) {
                printf("\nEnter a 50 character comment line:\n");
                printf("(Use _ instead of blank spaces!)\n");
                for(i = 1; i <= 5; i++) printf("1234567890");
                printf("\n");
                for(i = 1; i <= 6; i++) printf("%d         ", i-1);
                printf("\n");
                for(i = 0; i <= NDUMMY; i++) dummy[i] = '\0';
                strcpy(dummy,"ensemble-averaged velocity: ");
                printf("%s", dummy);
                scanf("%s", buffer);
                strcat(dummy,buffer);
                printf("Comment line: \n%s", dummy);
                printf("\n Entry correct? (y or n)\n");
                scanf("%s", resp);
                if(resp[0] == 'y' || resp[0] == 'Y')
                        {inerr = 0;}
                else if(resp[0] == 'n' || resp[0] == 'N')
                        {inerr = 1;}
                else
                        {printf(" Respond with y, Y for 'yes' ");
                        printf("or with n, N for 'no' next time.\n");}
                }

        /* Enter room air conditions. */
        if(! old_file) air_statex();

        /* Enter probe position. */
```

```
for(inerr = 1; inerr; ){
        printf("\nradial probe location:\n");
        printf("Code to enter wall distance in inches = '1'\n");
        printf("Code to enter wall distance in mm = '2'\n");
        printf("Code to enter radius as fraction ");
        printf("of the tube radius = '3'\n");
        printf("Code to quit = '0'\n");
        printf(" Enter code for entry.\n");
        while(scanf("%d", &irad) == 0)
                {getchar();
                printf(" Enter an integer code number!\n");}
        switch (irad)
        {
        case 0:
                {goto the_end;
                break;}
        case 1:
                {printf(" Enter wall distance in inches.\n");
                scanf("%lf", &dist);
                distance = dist * 0.0254;
                break;}
        case 2:
                {printf(" Enter wall distance in mm.\n");
                scanf("%lf", &dist);
                distance = dist / 1000.;
                break;}
        case 3:
                {printf(" Enter radius as a fraction of the");
                printf(" tube radius.\n");
                scanf("%lf", &relrad);
                distance = (1. - relrad) * 0.5 * mdiam;
                break;}
        }
        printf(" Wall distance =        %7.4g inches\n",
                                        distance / 0.0254);
        printf("                =       %6.3g mm\n",
                                        distance * 1000.);
        printf(" Radius / Tube Radius = %6.3g \n",
                                1. - 2.*distance/mdiam);
        printf(" Entry correct? (y or n)\n");
        printf(" Enter 'y' to start acquisition.\n");
        scanf("%s", resp);
        if(resp[0] == 'y' || resp[0] == 'Y')
                {inerr = 0;}
        else if(resp[0] == 'n' || resp[0] == 'N')
                {inerr = 1;}
        else
                {printf(" Respond with y, Y for 'yes' ");
                printf("or with n, N for 'no' next time.\n");}
        }

/* If no instrument is available on the IEEE interface ... */
if(no_inst) goto the_end;

/*      ********************************************************        */
/*              Acquire data with NORLAND until the                     */
/*              maximum number of cycles is reached                     */

printf(" Data will be acquired now.\n");

/* Set flag for acquisition of first round. */
first_time = 1;

/* Acquire and process as many rounds of data as requested. */
for(round = 1, next_round = 1; next_round; round++){

        /* Calculate density transient. */
        ref_dens = ref_press * air_mwt / gas_const / ref_temp;
        if((densfile = fopen("densfile","r")) != NULL) {
                fseek(densfile,0L,0);
                for(i = 0; i <= nread; i++) {
                        if(! fread(&density,sizeof(double),1,densfile)
```

```
                                        printf("Reading of density failed.\n")
                                umean[i] = ref_dens / density;
                                }
                        fclose(densfile);
                        }
                else {
                        printf("densfile does not exist. p_atm assumed\n");
                        mean_dens = p_atm * air_mwt / gas_const / t_dry;
                        for(i = 0; i <= nread; i++) {
                                umean[i] = ref_dens / mean_dens;
                                }
                        }

                /* Send string to start acquisition. */
                ibwrt(ib3,acquire,strlen(acquire));

                /* Wait until NORLAND starts acquisition. */
                /* (Loop to test whether NORLAND is already acquiring). */
                do{
                        ibrsp(ib3,&status_byte);
                        if(diagnosis)printf("status_byte = %o\n", status_byte)
                        }
                while(!status_byte & 5);
                if(diagnosis)printf("N started acquisition\n");

                /* If this is the first acquisition at this radial position,
                do not attempt to process data. */
                if(!first_time){
                        if(diagnosis)printf("This is not the first time.\n");

/*      ************************************************************      */
/*              loop through for both ueffA and ueffB,
                the effective cooling velocities                          */

                     for(ucount = 1; ucount<=2; ucount++){

                        if(ucount == 1) {
                                        ncalib = ncalibA;
                                        ptrdstr = readstrA;
                                        acalib = acalibA;
                                        bcalib = bcalibA;
                                        voffset = voffsetA;
                                        vgain = vgainA;}
                        else
                                        {
                                        ncalib = ncalibB;
                                        ptrdstr = readstrB;
                                        acalib = acalibB;
                                        bcalib = bcalibB;
                                        voffset = voffsetB;
                                        vgain = vgainB;}

                        /* Evaluate Factor and Offset from data sent in
                        XFAST binary format.  See NORLAND Prowler manual
                        volume 2 "Options for the Prowler", pp.46-48. */

                        /* Calculate Factor. */

                        /* The first two hex digits (Word 1) represent
                                the log to the base two,
                                biased by 128, of the first factor
                                of the Factor. */
                                sscanf(ptrdstr,"%2X",&ifactor1);

                        /* Compute the first factor of the Factor. */
                        factor1 = pow(2.,((double)ifactor1 - 128.));

                        /* The next six hex digits (Words 2, 3, 4) represent
                        the sign and the base two fractions of the second
                        factor of Factor.  Calculate contributions to the
                        second factor, hex-digit by hex-digit. */
                        for(i=0, factor2=0; i<6; i++){
                                strncpy(digit,(ptrdstr+2+i),1);
```

```
        digit[1] = '\0';

        /* Determine bit pattern corresponding to each
        hex-digit and calculate the second factor. */
        sscanf(digit,"%X",&hexdigit);
        bit0 = hexdigit & 1;
        bit1 = hexdigit & 2;
        bit2 = hexdigit & 4;
        bit3 = hexdigit & 8;
        if(i==0){
                if(bit3)
                        sign = -1;
                else
                        sign = 1;
                bit3 = 1;
                }
        if(bit3) factor2 = factor2
                        + pow(2.,-(double)(i*4 + 1));
        if(bit2) factor2 = factor2
                        + pow(2.,-(double)(i*4 + 2));
        if(bit1) factor2 = factor2
                        + pow(2.,-(double)(i*4 + 3));
        if(bit0) factor2 = factor2
                        + pow(2.,-(double)(i*4 + 4));
        }
factor = (double) sign * factor1 * factor2;
/* if(diagnosis)printf("factor = %f\n",factor); */

/* Calculate Offset. */

/* The first two hex digits
        (Word 1) represent the log to the base two,
        biased by 128, of the first factor
        of the Offset. */
sscanf((ptrdstr+8),"%2X",&ifactor1);

/* Compute the first factor of the Offset. */
factor1 = pow(2.,((double)ifactor1 - 128.));

/* The next six hex digits (Words 2, 3, 4) represent
the sign and the base two fractions of the second
factor of Offset. */
/* Calculate contributions to the second factor,
hex-digit by hex-digit. */
for(i=0, factor2=0; i<6; i++){
        strncpy(digit,(ptrdstr+10+i),1);
        digit[1] = '\0';

        /* Determine bit pattern corresponding to each
        hex-digit and calculate the second factor. */
        sscanf(digit,"%X",&hexdigit);
        bit0 = hexdigit & 1;
        bit1 = hexdigit & 2;
        bit2 = hexdigit & 4;
        bit3 = hexdigit & 8;
        if(i==0){
                if(bit3)
                        sign = -1;
                else
                        sign = 1;
                bit3 = 1;
                }
        if(bit3) factor2 = factor2
                        + pow(2.,-(double)(i*4 + 1));
        if(bit2) factor2 = factor2
                        + pow(2.,-(double)(i*4 + 2));
        if(bit1) factor2 = factor2
                        + pow(2.,-(double)(i*4 + 3));
        if(bit0) factor2 = factor2
                        + pow(2.,-(double)(i*4 + 4));
        }
offset = (double) sign * factor1 * factor2;
if(diagnosis)printf("offset = %f\n",offset);
```

```
        if(diagnosis)printf("fraction and offset are calc\n");

        /* ****************************** */
        /* Evaluate individual data points */

        icount = (MREAD2 - 1) / nread;
        ninv = 1. / ncalib;
        off_set = offset + voffset * vgain;
        volt_corr = sqrt((t_sensor - ref_temp)
                        /(t_sensor - t_dry)) / vgain;
        angle = PI * theta / 180.;
        sint = sin(angle);
        cost = cos(angle);
        jcount = 0;
for(i = 1, iptr = (int *)(ptrdstr+256);
    iptr < (int *)(ptrdstr+256+2*cy_in_buf*(MREAD2-1));
    i++, iptr++){

        /* NOTE:  Due to storage limitations, only every
        fourth point (2 degrees) is included, i.e.
        2 4 6 ... 180 182 ... 360 degrees */
        if(!(i % icount)){
                jcount++;
                /* Calculate voltage value according to
                NORLAND Prowler manual pp.48-49 of 5/20/85. */
                voltage = ((double)(*iptr-0X8000)*factor
                          + off_set) * volt_corr;

                /* Calculate instantaneous velocity. */
                /* King's Law with exponent other than 0.5 */
                if((base = voltage*voltage - acalib) > 0.0) {
                        veleff = pow(base/bcalib,ninv);
                        }
                else {
                        veleff = 0.0;
                        }

                /* Correct velocity for static pressure. */
/* Note: Currently, the density transient for the pressure
correction is corrected for angular offset between
piston TDC and TDC-marker in def_density; this also applies
to acq_u.c. 4/9/89 JS */
        veleff *= umean[jcount % (MREAD-1)];

        if(ucount == 1) ueffA[jcount] = veleff;
        else
        {
                ueffB = veleff;

                lhsA = pow(ueffA[jcount],2.);
                lhsB = pow(ueffB,2.);
                v = sint * (ueffA[jcount] - ueffB);
                u = (ueffA[jcount] - v * sint) / cost;
                iter = 0;

        /* ******************************************** */
        /* iteration to determine u and v instantaneous */

                do
                {
                iter++;
                ktAterm = pow((kt * sint * (u - v)),2.);
                ktBterm = pow((kt * cost * (u + v)),2.);
                ueffAnew = sqrt(lhsA - ktAterm);
                ueffBnew = sqrt(lhsB - ktBterm);
                vnew = sint * (ueffAnew - ueffBnew);
                unew = (ueffAnew - vnew * sint) / cost;
                epsilonA = fabs((vnew - v) / v);
                epsilonB = fabs((unew - u) / u);
                u = unew;
                v = vnew;
                if(iter > 2)
                printf("i = %d iter = %d epA = %f epB = %f\n",
```

```
                                        i,iter,epsilonA,epsilonB);
                                }
                                while  ((epsilonA > .001) || (epsilonB > .001));

                                iangle = jcount % (MREAD-1);

                                /* Update the sumations */
                                sumu[iangle] += u;
                                sumv[iangle] += v;
                                sumu2[iangle] += (u*u);
                                sumv2[iangle] += (v*v);
                                sumuv[iangle] += (u*v);

/* Output v instantaneous to the file "vminst" */
if((iangle == 179) && (i < 750)){
        vmfile = fopen("/usr/geoff/proc/vminst","w");
        for(kcount = 1; kcount <= 179; kcount++){
        fprintf(vmfile,"%d\t%lf\n",2 * kcount,sumv[kcount]);
        }
        fflush(vmfile);
        fclose(vmfile);
        printf("vminst file printed\n");

}
                }       /* end of loop for ueffB */
                }       /* end of loop for every 4th pt */
        }               /* end of binary string breakdown loop */
        }               /* end of ucount loop for ueffA and ueffB */

                if(diagnosis)printf("data converted\n");

        /* ******************************** */
        /* Section to determine run status */
                /* Should another buffer of data be acquired? */
                if((lastcy = (round - 1) * cy_in_buf) >= mcycle)
                                                next_round = 0;
                if(diagnosis) printf("next_round = %d\n",
                                                next_round);
                }
        if(diagnosis)printf("wait for end of acquisition\n");
        if(diagnosis)printf("status_byte = %o\n", status_byte);

        /* Send message about current round. */
        printf(" Data for round %d (cycle %d) processed.\n",
                (round - 1), (round - 1) * cy_in_buf);

        /* Wait until NORLAND is done acquiring. */
        /* (Loop to test whether NORLAND is still acquiring.) */
        i = 0;
        do{
                ibrsp(ib3,&status_byte);
                if(diagnosis){
                        i += 1;
                        if(i > 100){
                                i = 0;
                                printf("%o\n", status_byte);
                                }
                        }
                }
        while(status_byte & 5);
        if(diagnosis)printf("N stopped acquisition\n");
        if(diagnosis)printf("status_byte = %o\n", status_byte);

        /* If necessary, read results of data acquisition. */
        if(next_round){
        if(diagnosis)printf("reading of A will begin\n");
                /* Send command to read NORLAND buffer for Chan A. */
                ibwrt(ib3,rdcmdA,strlen(rdcmdA));
                /* Read data from NORLAND. */
                ibrd(ib3,readstrA,SMAX);
        if(diagnosis)printf("reading of A done\n");
        if(diagnosis)printf("status_byte = %o\n", status_byte);
```

```
if(diagnosis)printf("reading of B will begin\n");
        /* Send command to read NORLAND buffer for Chan B. */
        ibwrt(ib3,rdcmdB,strlen(rdcmdB));
        /* Read data from NORLAND. */
        ibrd(ib3,readstrB,SMAX);
if(diagnosis)printf("reading of B done\n");
        }
if(diagnosis)printf("status_byte = %o\n", status_byte);

/* Reset flag for first round of acquisition. */
if(first_time) first_time = 0;

/* Calculate the number of cycles stored and entered
   into the summation so far. */
cycles = (double)((round - 1) * cy_in_buf);
printf("cycles stored, summed = %f\n",cycles);


/* Set flags for printout and for updating sums. */
if(!next_round) {
        fin_prt = 1;
        }
if(m_out >= 1 && iout <= m_out
   && cycles >= (double)n_out[iout]) {
        iout++;
        int_prt = 1;
        }

if(fin_prt || int_prt) {
        printf("cycles = %f\n",cycles);

        /* ************************************** */
        /* Calculate ensemble averaged velocities,
        rms velocity fluctuations, and shear stresses
        based on the summations */

        for(iangle = 1; iangle <= nread; iangle++){
             um = sumu[iangle] / cycles;
             umean[iangle] = um;
             vm = sumv[iangle] / cycles;
             vmean[iangle] = vm;
             if((argument = 1./(cycles - 1.) *
                        (sumu2[iangle] - um*um*cycles)) > 0.0)
                        urms[iangle] = sqrt(argument);
                  else urms[iangle] = 0.0;
             if((argument = 1.0/(cycles - 1.) *
                        (sumv2[iangle] - vm*vm*cycles)) > 0.0)
                        vrms[iangle] = sqrt(argument);
                  else vrms[iangle] = 0.0;
             upvp[iangle] = 1.0/(cycles - 1.) *
                        (sumuv[iangle] - um*vm*cycles);
                }
        }

if(int_prt) {
        /* Store sums in auxiliary arrays to be updated
           further during subsequent rounds. */
        for(iangle = 1; iangle <= nread; iangle++) {
                old_su[iangle] = sumu[iangle];
                old_sv[iangle] = sumv[iangle];
                old_su2[iangle] = sumu2[iangle];
                old_sv2[iangle] = sumv2[iangle];
                old_suv[iangle] = sumuv[iangle];
                }
        }
if(fin_prt || int_prt) {
        if(diagnosis) printf("fin_prt = %d, int_prt = %d \n",
                                fin_prt, int_prt);

        /* Sort entries in umean and urms arrays according to
           crank angle to account for angular offset
           between TDC and trigger. */
```

```
for(iangle = 1; iangle <= nread - nlag; iangle++){
        sumu[iangle + nlag] = umean[iangle];
        sumv[iangle + nlag] = vmean[iangle];
        sumu2[iangle + nlag] = urms[iangle];
        sumv2[iangle + nlag] = vrms[iangle];
        sumuv[iangle + nlag] = upvp[iangle];
        }
for(iangle = nread - nlag + 1;
    iangle <= nread; iangle++){
        sumu[iangle + nlag - nread] = umean[iangle];
        sumv[iangle + nlag - nread] = vmean[iangle];
        sumu2[iangle + nlag - nread] = urms[iangle];
        sumv2[iangle + nlag - nread] = vrms[iangle];
        sumuv[iangle + nlag - nread] = upvp[iangle];
        }

/* ********************************** */
/* Store the results in a binary string */

if(diagnosis) printf("will open storefile\n");
storefile = fopen(filename,"r+");
if(diagnosis) printf("opened storefile\n");
if(storefile == NULL)
        printf("fopen failed\n");
/* Move to end of file. */
if(fseek(storefile,0L,2))
        printf("fseek failed\n");
if(fwrite(&distance,sizeof(double),1,storefile) != 1)
        printf("fwrite failed\n");
        if(diagnosis)printf("distance = %f\n",distance);
if(fwrite(&accuracy,sizeof(double),1,storefile) != 1)
        printf("fwrite failed\n");
        if(diagnosis)printf("accuracy = %f\n",accuracy);
if(fwrite(&lastcy,sizeof(int),1,storefile) != 1)
        printf("fwrite failed\n");
        if(diagnosis)printf("lastcy = %d\n",lastcy);
if(fwrite(&maxuerr,sizeof(double),1,storefile) != 1)
        printf("fwrite failed\n");
        if(diagnosis)printf("maxuerr = %f\n",maxuerr);
if(fwrite(&iuerr,sizeof(int),1,storefile) != 1)
        printf("fwrite failed\n");
        if(diagnosis)printf("iuerr = %d\n",iuerr);
if(fwrite(&mxrmserr,sizeof(double),1,storefile) != 1)
        printf("fwrite failed\n");
        if(diagnosis)printf("mxrmserr = %f\n",mxrmserr);
if(fwrite(&irmserr,sizeof(int),1,storefile) != 1)
        printf("fwrite failed\n");
        if(diagnosis)printf("irmserr = %d\n",irmserr);
if(fwrite(&sumu[1],sizeof(double),nread,storefile)
   != nread)
        printf("fwrite failed\n");
        if(diagnosis) {
         for(count3 = 1; count3 <= 180; count3++);
         printf("umean[%d] = %f\n",count3,sumu[count3]);
                        }
if(fwrite(&sumu2[1],sizeof(double),nread,storefile)
   != nread)
        printf("fwrite for sumu2 failed\n");
if(fwrite(&sumv[1],sizeof(double),nread,storefile)
   != nread)
        printf("fwrite for sumv failed\n");
if(fwrite(&sumv2[1],sizeof(double),nread,storefile)
   != nread)
        printf("fwrite for sumv2 failed\n");
if(fwrite(&sumuv[1],sizeof(double),nread,storefile)
   != nread)
        printf("fwrite for sumuv failed\n");
if(! old_file) {
        if(fwrite(&t_dry,sizeof(double),1,storefile)
           != 1) printf("fwrite for t_dry failed\n");
        if(fwrite(&t_wet,sizeof(double),1,storefile)
           != 1) printf("fwrite for t_wet failed\n");
        if(fwrite(&p_atm,sizeof(double),1,storefile)
```

```
                                != 1) printf("fwrite for p_atm failed\n");
                            if(fwrite(dummy,sizeof(char),NDUMMY,storefile)
                                != NDUMMY) printf("fwrite for NDUMMY failed\n");
                            if(diagnosis)printf("Comment: %s\n", dummy);
                            }
                        if(diagnosis) printf("will close storefile\n");
                        if(fclose(storefile) == EOF) printf("fclose failed\n");
                        printf("Data were stored in file.\n");
                        }

                    if(fin_prt) {
                            /* Reset the sums used for averaging to zero for the
                               next calculation of ensemble averages. */
                            for(iangle = 1; iangle <= nread; iangle++){
                                    sumu[iangle] = 0.0;
                                    sumv[iangle] = 0.0;
                                    sumu2[iangle] = 0.0;
                                    sumv2[iangle] = 0.0;
                                    sumuv[iangle] = 0.0;
                                    }
                            iout = 1;

                            /* Reset fin_prt to zero. */
                            fin_prt = 0;
                            }

                    if(int_prt) {
                            /* Reset the sums used for averaging to zero for the
                               ongoing calculation of ensemble averages. */
                            for(iangle = 1; iangle <= nread; iangle++){
                                    sumu[iangle] = old_su[iangle];
                                    sumv[iangle] = old_sv[iangle];
                                    sumu2[iangle] = old_su2[iangle];
                                    sumv2[iangle] = old_sv2[iangle];
                                    sumuv[iangle] = old_suv[iangle];
                                    }

                            /* Reset int_prt to zero. */
                            int_prt = 0;
                            }
                    }
            }

the_end: printf("No further data will be acquired.\n");
/*      Acquirex program end    */
/*      ********************    */
}
```

AIR_STATEX

```
air_statex()
/* Function to enter, or read from existing file, the ambient air
        conditions and to enter mean static air pressure in a file.
        Function adapted from air_state.c for cross-wires 3/90 */

{
#include <stdio.h>
#include <string.h>

#define  STRLNG 30
#define  NDUMMY 50

FILE  *storefile;

extern int  diagnosis, oldfile;
extern int  nread; /* number of readings per cycle */
extern char  filename[];
extern double  t_dry, t_wet, p_atm; /* air conditions during run,
                        stored in Kelvin and Pascal internally. */

int     inerr, /* error in input data */
        i /* auxiliary counter */
        ;
long  offset;
long  sizeofheader, sizeofset;
double  gas_const = 8315.,  /* universal gas constant in kg/kmol/K */
        air_mwt = 28.96,  /* molecular weight in kg/kmol for air
                             with mole fractions of:
                                N2 - 0.7809
                                O2 - 0.2095
                                Ar - 0.0096     */
        density  /* air density in kg/m^3 */
        ;
char  resp[STRLNG];

        /* Open storefile. */
        storefile = fopen(filename,"r");

        /* Enter ambient air conditions. */
        if(diagnosis) printf("oldfile = %d\n", oldfile);
        if(oldfile) {
                printf("Current ambient air conditions:\n");
                resp[0] = 'n';
                }
        else {
                printf("\nEnter ambient air conditions.\n");
                resp[0] = 'y';
                oldfile = 1;
                }
        switch(resp[0]) {
        case 'n': case 'N':{
                /* Determine the size of the header. */
                sizeofheader = (long)((10+NDUMMY)*sizeof(char)
                                      + 6*sizeof(double)
                                      + 2*sizeof(int));
                if(diagnosis) printf("sizeofheader = %ld\n", sizeofheader);
                /* Determine the size of a data set. */
                sizeofset = (long)(7*sizeof(double) + 3*sizeof(int)
                                + 2 * nread * sizeof(double)
                                + NDUMMY * sizeof(char));

                /* Addition to size of a data set for cross wire */
                sizeofset += (long)(3 * nread * sizeof(double));

                if(diagnosis) printf("sizeofset = %ld\n", sizeofset);

                /* Determine offset of latest air data
                                                from end of file. */
                fseek(storefile,0L,2); /* Move to end of file. */
```

```
        if(diagnosis) printf("fseek @ end ok\n");

        /* Check whether there is an entry for the state
                of the air.  Read the air data. */
        offset = ftell(storefile);
        if(diagnosis) printf("offset = %ld\n", offset);
        if(offset >= sizeofheader + sizeofset) {
                if(diagnosis) printf("ftell >\n");
                offset = (long)(- 3 * sizeof(double)
                                - NDUMMY * sizeof(char));
                fseek(storefile,offset,2);
                offset = ftell(storefile);
                if(diagnosis) {
                        printf("offset = %ld\n", offset);
                        printf("errors before reading: ");
                        if(feof(storefile)) printf("EOF\n");
                        if(ferror(storefile)) printf("error\n");
                        }
                if(fread(&t_dry,sizeof(double),1,storefile) != 1) {
                        if(feof(storefile)) printf("EOF\n");
                        if(ferror(storefile)) printf("error\n");
                        }
                else {
                        if(diagnosis) printf("fread of t_dry succ.\n");
                        }
                if(fread(&t_wet,sizeof(double),1,storefile) != 1) {
                        if(feof(storefile)) printf("EOF\n");
                        if(ferror(storefile)) printf("error\n");
                        }
                else {
                        if(diagnosis) printf("fread of t_wet succ.\n");
                        }
                if(fread(&p_atm,sizeof(double),1,storefile) != 1) {
                        if(feof(storefile)) printf("EOF\n");
                        if(ferror(storefile)) printf("error\n");
                        }
                else {
                        if(diagnosis) printf("fread of p_atm succ.\n");
                        }
                printf("dry-bulb temperature = %lf degr.C\n",
                                        t_dry - 273.15);
                printf("wet-bulb temperature = %lf degr.C\n",
                                        t_wet - 273.15);
                printf("atmospheric pressure = %lf bar\n",
                                        p_atm / 1.e+05);
                printf("Entry correct? (y/n)");
                scanf("%s", resp);
                if(resp[0] == 'y' || resp[0] == 'Y')
                        break;
                }
        else {
                printf("There is no old entry of ambient air data.\n")
                }
        }
case 'y': case 'Y':{
        /* Enter ambient air conditions. */
        for(inerr = 1; inerr; ){
                printf("Enter\tdry-bulb temperature (degr.C),\n");
                printf("\twet-bulb temperature (degr.C),\n");
                printf("\tatmospheric pressure (bar).\n");
                scanf("%lf%lf%lf", &t_dry, &t_wet, &p_atm);
                printf("dry-bulb temperature = %lf degr.C\n",
                                t_dry);
                printf("wet-bulb temperature = %lf degr.C\n",
                                t_wet);
                printf("atmospheric pressure = %lf bar\n",
                                p_atm);
                t_dry += 273.15;
                t_wet += 273.15;
                p_atm *= 1.e+05;
                printf("Entry correct? (y/n)\n");
                scanf("%s", resp);
                if(resp[0] == 'y' || resp[0] == 'Y')
```

```
                        inerr = 0;
                    else if(resp[0] == 'n' || resp[0] == 'N')
                        inerr = 1;
                    else {
                        printf("Enter numbers and \n");
                        printf("respond with y, Y for 'yes' ");
                        printf("or with n, N for 'no' next time.
                                    \n");
                        }
                    }
                }
            break;
            }
            fclose(storefile);
}
```

PROCESSX

```
/* "processx.c" processes two channel xwire measurement files.
        G. Friedman, 8/90 */
/* Program is adapted from "process.c" written by J. Seume for single
     - wire data processing */

#include <stdio.h>
#include <math.h>
#include <string.h>

#define MRAD 40
#define MREAD 181
#define IDLENGTH 10
#define NDUMMY 50
#define PI (4. * atan(1.0))
#define NU 16.e-06

char  usage[] = "Usage: procshw [-f runid or -t position or -p time or\
-v or -a or -d or -w or -r or -R]\n";

main(argc,argv)
int  argc;
char  *argv[];
{
FILE  *storefile;
FILE  *tracefile;
FILE  *proffile;
FILE  *avefile;
FILE  *idealfile;

int  write_all;          /* Write to files the data for all active traces and
                         for profiles in increments of 30 degs of crank angle. */
int  diagnosis = 1;      /* Print error diagnostics. */
int  fast = 1;           /* Enter file i.d. from the command line,
                         skip echo print of run parameters. */
int radial = 1;          /* Process the radial velocity component v */
int  trace = 1;          /* Generate data for a plot of the ensemble-averaged
                         mean-velocity fluctuation transient
                         at one radial location. */
int  profile = 1;        /* Generate data for a plot of a mean-vel. profile */
int  average = 1;        /* Generate data for a plot of the cross-sectional
                         mean mean-velocity transient (real and ideal). */
int  rms = 1;            /* Process rms-velocity fluctuations instead of
                         ensemble-averaged velocities. */
int  Rss = 1;            /* Process Reynolds' shear stress values */
int  position;           /* Number of the radial position at which a trace shall
                         be plotted. */
int  itime;              /* Number of the current crank angle. */
int  not_done = 1;       /* Program is not ready to be terminated. */
int  old_file = 0;       /* file format old or new ? */
int  inerr, flag, iangle, nrad, nsort, i, j, irad;
int  nread;              /* number of readings per cycle */
int  mcycle;             /* maximum number of cycles to be acquired */
int  nangle = 12;        /* number of angles at which profiles are to be plotted
                         when write_all = 1 */
int  lastcy, iuerr, irmserr;
long  p, rptr[MRAD], offset;
long  sizeofheader, sizeofset;
double  angle;           /* crank angle */
double  time;            /* Integer crank angle at which a mean-velocity profil
                         shall be plotted. */
double  mstroke, mbore, mdiam, mlength, maxial, freq;
double  relrad;
double  Remax, Va, Ar, loverd, xoverl, xoverd, uavemax;
double  accuracy, maxuerr, mxrmserr;
static double  distance, umean[MREAD], urms[MREAD], uave[MREAD];
static double  vmean[MREAD], vrms[MREAD], vave[MREAD], upvp[MREAD];
static double  dist[MRAD], value, umprf[MRAD], ucl;
double  uprt;            /* auxiliary variable to print umean, urms */
double  d, sortdist[MRAD];
```

```
double  umint, aint;
double  rim1, ri;       /* auxiliary radii */
double  fnull = 0.0;
double  pi; /* pi = 3.1415.... */
double  air_viscosity(), air_cond(), air_cp(),
        air_density(), dry_air_density(), psat_water();
double  T_dry, T_wet, p_atm;
double  rho, rhod, hum_ratio, air_cont, rh, mu, nu, cond, cp, alpha, Pr;

char  runid[10], resp[20],resp2[20], filename[30], title[NDUMMY], dummy[NDUMMY
char  tfname[10], tfnumber[5], pfname[10], pfnumber[5];
char  year[3], month[3];

/* Calculate pi */
pi = PI;
printf("nread = %d\n",nread);

        /* Determine the sizes of header and data sets. */
        sizeofheader = (long)(10*sizeof(char) + 6*sizeof(double))
                        + sizeof(int);  /* size of the run information */
        sizeofset = (long)(4*sizeof(double) + 3*sizeof(int)
                        + 2 * nread * sizeof(double)); /* size of one
                                                          data set */
        /* Add to "sizeofset" the size of the 3 new arrays used in
        the cross-wire measurements */
        sizeofset += (long)(3 * nread * sizeof(double));

/* Set defaults for command line parameters. */
fast = 0;
radial = 0;
trace = 0;
profile = 0;
average = 0;
diagnosis = 0;
write_all = 0;
rms = 0;
Rss = 0;

/* Read the command line for control parameters. */
while(*++*argv == '-'){
        --argc;
        switch(*++*argv)
        {
        case 'f':
                fast = 1;
                strcpy(runid,*++argv);
                --argc;
                break;
        case 't':
                trace = 1;
                position = atoi(*++argv);
                --argc;
                break;
        case 'v':
                radial = 1;
                --argc;
                break;
        case 'p':
                profile = 1;
                time = atof(*++argv);
                --argc;
                break;
        case 'a':
                average = 1;
                --argc;
                break;
        case 'd':
                diagnosis = 1;
                --argc;
                break;
        case 'w':
                write_all = 1;
                --argc;
```

```
                break;
        case 'r':
                rms = 1;
                --argc;
                break;
        case 'R':
                Rss = 1;
                --argc;
                break;
        }
        }

if(diagnosis)printf("trace position = %d\n",position);
for(not_done = 1; not_done; ){
        if(! fast){
                /* Read run identification and check whether the corresponding
                file already exists. */
                for(inerr = 1; inerr; ){
                        printf("\nEnter run identification: \n\n");
                        printf("(Use the format mmddyyss where: \n");
                        printf(" mm = month, dd = day, yy = year,\n");
                        printf(" and ss = a sequence number of the day's");
                        printf(" runs)\n");
                        printf("Enter c to continue using the same ");
                        printf("run.\n");
                        printf("Enter s to stop the program.\n");
                        scanf("%s", runid);
                        if(runid[0] == 'c') break;
                        if(runid[0] == 's') goto the_end;
                        printf("mm = %c%c    ", runid[0], runid[1]);
                        printf("dd = %c%c    ", runid[2], runid[3]);
                        printf("yy = %c%c    ", runid[4], runid[5]);
                        printf("ss = %c%c\n ", runid[6], runid[7]);
                        strcpy(filename,"/usr/geoff/shwdata/");
                        strcpy(&filename[19],runid);
                        printf(" The filename is '%s'\n\n", filename);
                        fclose(storefile);
                        storefile = fopen(filename, "r+");
                        if(storefile == NULL)
                                printf(" This file does not exist.\n ");
                        else
                                printf(" This file exists.\n");
                        printf("\n Entry correct? (y or n or stop)\n");
                        scanf("%s", resp);
                        if(resp[0] == 'y' || resp[0] == 'Y')
                                {inerr = 0;}
                        else if(resp[0] == 'n' || resp[0] == 'N')
                                {continue;}
                        else if(resp[0] == 's' || resp[0] == 'S')
                                {goto the_end;}
                        else
                                {printf(" Respond with y, Y for 'yes'; ");
                                printf("or with n, N for 'no';\n");
                                printf("or with s, S for 'stop' next time.\n");}
                        }
                printf("If you want to process instantaneous ");
                printf("or ensemble-averaged data or times,\n");
                printf("=> enter 'm',\n");
                printf("If you want to process differences ");
                printf("or rms-fluctuations or angular velocity,\n");
                printf("=> enter 'r',\n");
                printf("If you want to process Reynolds' ");
                printf("shear stress,\n");
                printf("=> enter 'R',\n");
                scanf("%s", resp);
                if(resp[0] == 'm' || resp[0] == 'r')
                    {printf("Enter a 'v' for radial velocity or any other ");
                     printf("character for the axial velocity,\n");
                     scanf("%s",resp2);
                        if(resp2[0] == 'v') radial = 1;
                     }
                if(resp[0] == 'm') rms = 0;
                else if(resp[0] == 'r') rms = 1;
```

```
        else if(resp[0] == 'R') Rss = 1;
        else{
                printf("Mean velocity will be processed.\n");
                Rss = 0;
                rms = 0;
                }
        }
else    {
        /* Take run i.d. from command line. */
        strcpy(filename,"/usr/geoff/shwdata/");
        strcpy(&filename[19],runid);
        storefile = fopen(filename, "r+");
        if(storefile == NULL){
                printf(" This file does not exist.\n ");
                goto the_end;
                }
        }

/* Read parameters from data file. */
storefile = fopen(filename, "r+");
fread(runid,sizeof(char),10,storefile);
fread(&mstroke,sizeof(double),1,storefile);
fread(&mbore,sizeof(double),1,storefile);
fread(&mdiam,sizeof(double),1,storefile);
fread(&mlength,sizeof(double),1,storefile);
fread(&maxial,sizeof(double),1,storefile);
fread(&freq,sizeof(double),1,storefile);
fread(&nread,sizeof(int),1,storefile);
/* Check whether this is a file of the old format. */
strncpy(year,&runid[4],2);
strncpy(month,&runid[0],2);
if(atoi(year) == 88 && atoi(month) < 6)
        old_file = 1;
else
        old_file = 0;
if(diagnosis)printf("old_file = %d\n",old_file);
if(! old_file) {
        fread(&mcycle,sizeof(int),1,storefile);
        fread(title,sizeof(char),NDUMMY,storefile);
        }
fseek(storefile,0L,0); /* rewind */

/* Determine the sizes of header and data sets. */
sizeofheader = (long)(10*sizeof(char) + 6*sizeof(double))
                + sizeof(int); /* size of the run information */
sizeofset = (long)(4*sizeof(double) + 3*sizeof(int)
                + 2 * nread * sizeof(double)); /* size of one
                                                  data set */

if(! old_file) {
        sizeofheader += (long)(sizeof(int) + NDUMMY * sizeof(char));
        sizeofset += (long)(3 * sizeof(double) + NDUMMY * sizeof(char));
        }
/* Add to "sizeofset" the size of the 3 new arrays used in
the cross-wire measurements */
sizeofset += (long)(3 * nread * sizeof(double));

/* Print title. */
printf("\n Title:\n");
printf("%s\n", title);

if(! fast){
        /* Print data summaries in English and SI units
        and dimensionless form. */
        /* Echo print input data in English units. */
        printf("\n Input data summary in English units:\n");
        printf(" stroke =       %6.1f in\n", mstroke/0.0254);
        printf(" bore   =       %6.1f in\n", mbore/0.0254);
        printf(" test section diameter =        %6.1f in\n",
                                                mdiam/0.0254);
        printf(" test section length   =        %6.1f in\n",
                                                mlength/0.0254);
        printf(" axial location        =        %6.1f in\n",
```

```
                                                maxial/0.0254);
printf(" drive shaft freq =     %6.1f rpm\n", freq*240.);
printf( " %d readings taken per cycle\n", nread);
if(! old_file) {
        printf(" Maximum number of cycles ");
        printf(" to be acquired = %d\n", mcycle);
        printf(" Comment: %s\n", dummy);
        }
printf("Type a character followed by <CR> to continue.\n");
if(scanf("%s", resp));

/* Echo print input data in SI units. */
printf("\n Input data summary in SI units:\n");
printf(" stroke =       %6.1f mm\n", mstroke*1000);
printf(" bore   =       %6.1f mm\n", mbore*1000);
printf(" test section diameter =        %6.1f mm\n",
                                        mdiam*1000);
printf(" test section length   =        %6.1f mm\n",
                                        mlength*1000);
printf(" axial location        =        %6.1f mm\n",
                                        maxial*1000);
printf(" frequency =    %6.3f Hz\n", freq);
printf( " %d readings taken per cycle\n", nread);

/* estimate of the amplitude of the bulk-mean velocity */
uavemax = PI * freq * mstroke * (mbore*mbore) / (mdiam*mdiam);
printf("Estimate of the amplitude of the bulk-mean velocity ");
printf("= %5.2f m/sec\n", uavemax);

/* Calculate and print similarity parameters. */
printf("\n Nominal similarity parameters:\n");
printf(" Remax =        %10.2e\n",
        Remax = PI*mbore*mbore*freq*mstroke / mdiam / NU);
printf(" Va =           %6.1f\n",
        Va = 0.5*PI*freq*mdiam*mdiam / NU);
printf(" Ar =           %6.2f\n",
        Ar = mbore*mbore/(mdiam*mdiam)*mstroke/mlength);
printf(" l/d =          %6.1f\n", loverd = mlength / mdiam);
printf(" x/l =          %6.3f\n", xoverl = maxial / mlength);
printf(" x/d =          %6.1f\n", xoverd = maxial / mdiam);

/* Reset control parameters. */
trace = 0;
profile = 0;
average = 0;

/* Enter type of output desired. */
for(inerr = 1; inerr; ){
        printf("\nEnter         t       for     trace\n");
        printf("              p               profile\n");
        printf("              a               average\n");
        scanf("%s", resp);
        switch(resp[0])
        {
        case 't':
                trace = 1;
                printf("Trace will be plotted.\n");
                break;
        case 'p':
                profile = 1;
                printf("Profile will be plotted.\n");
                break;
        case 'a':
                average = 1;
                printf("Transient of bulk-mean ");
                printf("quantity will be plotted.\n");
                break;
        default:
                printf("Enter correct code letter.\n");
                break;
        }

        printf("\n Entry correct? (y or n)\n");
```

```
                scanf("%s", resp);
                if(resp[0] == 'y' || resp[0] == 'Y')
                        inerr = 0;
                else if(resp[0] == 'n' || resp[0] == 'N'){
                        /* Reset control parameters. */
                        trace = 0;
                        profile = 0;
                        average = 0;
                        }
                else{
                        printf(" Respond with y, Y for 'yes'; ");
                        printf("or with n, N for 'no' next time.\n");
                        /* Reset control parameters. */
                        trace = 0;
                        profile = 0;
                        average = 0;
                        }
                }
        }

/* Scan the file for wall distance entries. */
fseek(storefile,sizeofheader,0); /* Rewind and move to
                beginning of first data set. */
flag = 1;
nrad = 0;
i = 1;
while(flag != 0){
        flag = fread(&dist[i],sizeof(double),1,storefile);
        if(flag)
                {nrad = nrad + 1;
                if(diagnosis) printf("Read set # %d\n", nrad);
                rptr[i] = ftell(storefile)
                                - (long) sizeof(double);
                fseek(storefile,sizeofset
                                - (long)sizeof(double),1);}
        else
                {if(diagnosis) printf("flag = 0\n");}
        i = i + 1;
        }
        if(diagnosis)printf("# of radial data sets = %d\n",nrad);

if(! fast){
        /* Echo list of radial data sets. */
        printf(" There are %d radial data sets:\n", nrad);
        printf(" (Negative distance indicates deactivated ");
        printf("data set.\n");
        printf("Number \tdistance \tdistance \tradius /\n");
        printf("       \t(mm)     \t(inches) \ttube ");
        printf("radius\n");
        for(i = 1; i <= nrad; i++) {
                printf("%d      %f      %f      %f\n",
                        i, dist[i]*1000., dist[i]/0.0254,
                                1. - 2.*dist[i]/mdiam);
                }

        /* To remove data sets, substitute distance from the
        wall by its negative value. */
        printf("\n Do you want to (de)activate any data set? ");
        printf("(y or n)\n");
        scanf("%s", resp);
        if(resp[0] == 'y' || resp[0] == 'Y'){
                printf("Enter numbers of sets to be ");
                printf("(de)activated. (Enter s to ");
                printf("terminate.)\n");
                while(scanf("%s", resp)){
                        if(resp[0] == 's') break;
                        i = atoi(resp);
                        dist[i] = - dist[i];
                        fseek(storefile,
                                sizeofset*(long)(i-1) +
                                        sizeofheader,0);
                        fwrite(&dist[i],sizeof(double),1,storefile);
                        }
```

```
                        }
                }

        if(trace){
                if(! fast){
                        /* Choose a radial location at which the traces are to
                        be plotted. */
                        printf("Enter number of the trace to be plotted.\n");
                        while(scanf("%d", &position) == 0){
                                getchar();
                                printf(" Enter the integer number ");
                                printf("of the trace!\n");
                                }
                        }
                /* Move to the data set of this radial location. */
                fseek(storefile,(long)(position-1)*sizeofset
                                + sizeofheader,0); /* Move to
                                beginning of desired data set. */
if(diagnosis)printf("sizeofset = %d\n",sizeofset);
if(diagnosis)printf("sizeofheader = %d\n",sizeofheader);

                /* Read in the desired data set. */
                fread(&distance,sizeof(double),1,storefile);
                fread(&accuracy,sizeof(double),1,storefile);
                fread(&lastcy,sizeof(int),1,storefile);
                fread(&maxuerr,sizeof(double),1,storefile);
                fread(&iuerr,sizeof(int),1,storefile);
                fread(&mxrmserr,sizeof(double),1,storefile);
                fread(&irmserr,sizeof(int),1,storefile);
                fread(&umean[1],sizeof(double),nread,storefile);
                fread(&urms[1],sizeof(double),nread,storefile);
                fread(&vmean[1],sizeof(double),nread,storefile);
                fread(&vrms[1],sizeof(double),nread,storefile);
                fread(&upvp[1],sizeof(double),nread,storefile);
                if(! old_file) {
                        fread(&T_dry,sizeof(double),1,storefile);
                        fread(&T_wet,sizeof(double),1,storefile);
                        fread(&p_atm,sizeof(double),1,storefile);
                        /* Read and print comment. */
                        fread(dummy,sizeof(char),NDUMMY,storefile);
                        printf("Comment: %s\n", dummy);
                        }

                /* Print out air state for this trace. */
                if(! old_file) {
                        printf("dry-bulb temperature = %lf degr.C\n",
                                                T_dry - 273.15);
                        printf("wet-bulb temperature = %lf degr.C\n",
                                                T_wet - 273.15);
                        printf("atmospheric pressure = %lf bar\n",
                                                p_atm * 1.e-05);
                        }

                /* Print out air properties for this trace. */
                if(! old_file && (! fast))
                 for(inerr = 1; inerr;) {
                        printf("Do you want to list property values? ");
                        printf("(y or n)\n");
                        scanf("%s", resp);
                        if(resp[0] == 'y' || resp[0] == 'Y') {
                                printf("Enter humidity ratio, air content ");
                                printf("(kg/m^3) \n");
                                printf("from psychrometric chart ");
                                printf("at 1 atmosphere.\n");
                                scanf("%lf%lf", &hum_ratio, &air_cont);
                                rhod = dry_air_density(T_dry, p_atm);
                                rho = (p_atm / 1.013e+05) * air_cont
                                        * (1. + hum_ratio);
                                printf("Air density (dry, humid) = %lg, %lg\n",
                                        rhod, rho);
```

```
                        printf("Air density decrease ");
                        printf("due to humidity = %lg%%\n",
                                (1. - rho / rhod) * 100.);
                        printf("Relative humidity = %lg%%\n",
                                rh = 100. * hum_ratio
                                * (p_atm - psat_water(T_dry))
                                / 0.622 / psat_water(T_dry));
                        printf("Air dynamic viscosity = %lg\n",
                                mu = air_viscosity(T_dry));
                        printf("Air kinematic viscosity = %lg\n",
                                nu = mu / rho);
                        printf("Air thermal conductivity = %lg\n",
                                cond = air_cond(T_dry));
                        printf("Air specific heat at constant ");
                        printf("pressure = %lg\n", cp = air_cp(T_dry));
                        printf("Air thermal diffusivity = %lg\n",
                                alpha = cond / rho / cp);
                        printf("Air Prandtl number = %lg\n",
                                Pr = nu / alpha);
                        printf("Entry correct? ");
                        printf("(y or n)\n");
                        scanf("%s", resp);
                        if(resp[0] == 'y' || resp[0] == 'Y') {
                                inerr = 0;
                                }
                        }
                else if(resp[0] == 'n' || resp[0] == 'N') {
                        inerr = 0;
                        }
                else {
                        printf(" Respond with y, Y for 'yes'; ");
                        printf("or with n, N for 'no';\n");
                        printf("or with s, S for 'stop' next time.\n");
                        }
                }

        /* Print data in column format to be plotted. */
        tracefile = fopen("trace","w");
        for(i = 1; i <= nread; i++){
            if(rms) { if(!radial)
                        { uprt = urms[i];
                        if(diagnosis)printf("processing urms\n");
                        }
                      else
                        { uprt = vrms[i];
                        if(diagnosis)printf("processing vrms\n");
                        }
                    }
             else if(Rss)
                        { uprt = upvp[i];
                        if(diagnosis)printf("processing upvp\n");
                        }
             else
                    {
                      if(!radial)
                        { uprt = umean[i];
                        if(diagnosis)printf("processing umean\n");
                        }
                          else
                        { uprt = vmean[i];
                        if(diagnosis)printf("processing vmean\n");
                        }
                    }
              fprintf(tracefile, "%f    %f\n",
                        (double) i * 360. / (double) nread, uprt);
         }
        fclose(tracefile);
        }

if(write_all){
        for(i = 1; i <= nrad; i++){
```

```
                        /* Read and process non-deactivated traces. */
                        if(dist[i] > 0.0){

                                /* Move to the data set of this radial
                                        location. */
                                fseek(storefile,(long)(i - 1)*sizeofset
                                        + sizeofheader,0); /* Move to
                                        beginning of desired data set. */

                                /* Read in the desired data set. */
                                fread(&distance,sizeof(double),1,storefile);
                                fread(&accuracy,sizeof(double),1,storefile);
                                fread(&lastcy,sizeof(int),1,storefile);
                                fread(&maxuerr,sizeof(double),1,storefile);
                                fread(&iuerr,sizeof(int),1,storefile);
                                fread(&mxrmserr,sizeof(double),1,storefile);
                                fread(&irmserr,sizeof(int),1,storefile);
                                fread(&umean[1],sizeof(double),nread,storefile);
                                fread(&urms[1],sizeof(double),nread,storefile);
                                fread(&vmean[1],sizeof(double),nread,storefile);
                                fread(&vrms[1],sizeof(double),nread,storefile);
                                fread(&upvp[1],sizeof(double),nread,storefile);

                                /* Print data in column format to be plotted. */
                                strcpy(tfname,"trace");
                                sprintf(tfnumber,"%d",i);
                                strcat(tfname,tfnumber);
                                tracefile = fopen(tfname,"w");
                                for(iangle = 1; iangle <= nread; iangle++){
                                  if(rms) { if(!radial) uprt = urms[iangle];
                                              else
                                                        uprt = vrms[iangle];
                                            }
                                  else if(Rss)          uprt = upvp[iangle];
                                  else    {
                                           if(!radial) uprt = umean[iangle];
                                           else
                                                       uprt = vmean[iangle];
                                          }
                                         fprintf(tracefile, "%f  %f\n",
                                         (double) iangle * 360. / (double) nread,
                                          uprt);
                                         }
                                fclose(tracefile);
                                }
                        }
                }

if(profile || average || write_all){
        /* Sort traces with increasing wall distance. */
        /* Write wall distances and data-set pointers into
                new arrays, omitting deactivated data sets. */
        for(i = 1, j = 1; i <= nrad; i++){
                if(dist[i] > 0.0){
                        sortdist[j] = dist[i];
                        rptr[j] = rptr[i];
                        j = j + 1;
                        }
                }
        nsort = j - 1;

        /* Rearrange the arrays. */
        /* (See section 8.1 of W. H. Press et al.: Numerical
                Recipes, Cambridge (UK) University Press 1986.) */
        for(j = 2; j <= nsort; j++){
                d = sortdist[j];
                p = rptr[j];
                for(i = j - 1; i >= 1; --i){
                        if(sortdist[i] > d){
                                sortdist[i+1]
                                        = sortdist[i];
                                rptr[i+1] = rptr[i];
                                }
```

```
                        else goto sortmark;
                        }
                i = 0;
sortmark:       sortdist[i+1] = d;
                rptr[i+1] = p;
                }

        if(diagnosis){
                printf("Sorting results:\n");
                printf("# \tdistance \tsorted \n");
                for(i = 0; i <= nrad; i++)
                        printf("%d \t%f \t%f \n", i, dist[i],
                                                sortdist[i]);
                }
        }

if(profile){
        if(! fast){
                /* Choose phase angle at which the profile is to
                be plotted. */
                printf("Enter the phase angle.\n");
                while(scanf("%lf", &time) == 0){
                        getchar();
                        printf(" Enter the integer ");
                        printf("phase angle!\n");
                        }
                }

        /* Calculate the offset from the beginning of the data
                set to the entry corresponding to the time of
                        interest in the storage file. */
        /* If it is for u */
        if(!radial && !Rss) offset = (long)(3 * sizeof(int))
                + (long)(4 * sizeof(double))
                + (long)(rms * nread * sizeof(double))
                + (long)((time * (double)nread / 360. - 1.)
                                                * sizeof(double));
        /* If it is for v */
        else if(radial) offset = (long)(3 * sizeof(int))
                + (long)(4 * sizeof(double))
                + (long)(2 * nread * sizeof(double))
                + (long)(rms * nread * sizeof(double))
                + (long)((time * (double)nread / 360. - 1.)
                                                * sizeof(double));
        /* If it is for upvp */
        else if(Rss) offset = (long)(3 * sizeof(int))
                + (long)(4 * sizeof(double))
                + (long)(4 * nread * sizeof(double))
                + (long)((time * (double)nread / 360. - 1.)
                                                * sizeof(double));

        if(diagnosis){
                printf("offset of entry = %ld", offset);
                printf("nread = %d", nread);
                printf("time = %f", time);
                printf("rms = %d", rms);
                printf("Rss = %d",Rss);
                printf("radial = %d",radial);
                }

        /* Read entries from data storage file and write to
                plot file. */
        proffile = fopen("proffile","w");
        fprintf(proffile,"%f    %g\n", 0.0, umprf[0] = 0.0);
        for(i = 1; i <= nsort; i++){
                /* Move to the entries corresponding
                                        to this phase angle. */
                fseek(storefile,(rptr[i] + offset),0); /* Move to the
                                        desired entry. */
                /* Read in the desired entry. */
                fread(&umprf[i],sizeof(double),1,storefile);
                if(diagnosis){
                        fseek(storefile,(rptr[i] + offset),0);
```

```
                        fread(&d,sizeof(double),1,storefile);
                        printf("radius =%f\n", d);
                        }

                /* Print data in column format to be plotted. */
                fprintf(proffile, "%f    %g\n",
                                        sortdist[i]/mdiam,
                                        umprf[i]);
                }
        fclose(proffile);
        }

if(write_all){
        for(iangle = 1; iangle <= nangle; iangle++){

                /* Choose phase angle at which the profile is to
                be plotted (every 30 degrees). */
                time = (360. / (double)nangle) * (double)iangle;

                /* Calculate the offset from the beginning of the data
                        set to the entry corresponding to the time of
                        interest in the storage file. */
                /* If it is for u */
                if(!radial && !Rss) offset = (long)(3 * sizeof(int))
                        + (long)(4 * sizeof(double))
                        + (long)(rms * nread * sizeof(double))
                        + (long)((time * (double)nread / 360. - 1.)
                                        * sizeof(double));
                /* If it is for v */
                else if(radial)
                     offset = (long)(3 * sizeof(int))
                        + (long)(4 * sizeof(double))
                        + (long)(2 * nread * sizeof(double))
                        + (long)(rms * nread * sizeof(double))
                        + (long)((time * (double)nread / 360. - 1.)
                                        * sizeof(double));
                /* If it is for upvp */
                else if(Rss)
                      offset = (long)(3 * sizeof(int))
                        + (long)(4 * sizeof(double))
                        + (long)(4 * nread * sizeof(double))
                        + (long)((time * (double)nread / 360. - 1.)
                                        * sizeof(double));

                /* Read entries from data storage file and write to
                        plot file. */
                strcpy(pfname,"profile");
                sprintf(pfnumber,"%d", (int)time);
                strcat(pfname,pfnumber);
                proffile = fopen(pfname,"w");
                fprintf(proffile,"%f    %f\n", 0.0, umprf[0] = 0.0);
                for(i = 1; i <= nsort; i++){
                        /* Move to the entries corresponding
                                                to this phase angle. */
                        fseek(storefile,(rptr[i] + offset),0); /* Move
                                        to the desired entry. */
                        if(diagnosis){
                                fread(&d,sizeof(double),1,storefile);
                                printf("radius = %f\n", d);
                                }

                        /* Read in the desired entry. */
                        fread(&umprf[i],sizeof(double),1,storefile);

                        /* Print data in column format to be plotted. */
                        fprintf(proffile, "%f    %g\n",
                                                sortdist[i]/mdiam,
                                                umprf[i]);
                        }
                fclose(proffile);
                }
        }
```

```
if(average){
        /* Open file for results of computation of the
        cross-sectional average mean velocity. */
        avefile = fopen("avefile","w");
        if(avefile == NULL) printf("avefile was not opened.");
        idealfile = fopen("idealfile","w");
        if(idealfile == NULL) printf("idealfile was not opened.");

        /* Calculate cross-sectional area for averaging. */
        /* The integral excludes areas beyond the centerline. */
        aint = mdiam * mdiam / 8.;

        /* Vary time. */
        for(itime = 1; itime <= nread; itime++){
                if(diagnosis) printf("itime =%d\n", itime);

                /* Calculate the offset from the beginning of the data
                        set to the entry corresponding to the time of
                                interest in the storage file. */

                /* If it is for u */
                if(!radial) offset = (long)(3 * sizeof(int))
                        + (long)(4 * sizeof(double))
                        + (long)(rms * nread * sizeof(double))
                        + (long)((time * (double)nread / 360. - 1.)
                                                * sizeof(double));
                /* otherwise it is for v */
                else
                        offset = (long)(3 * sizeof(int))
                        + (long)(4 * sizeof(double))
                        + (long)(2 * nread * sizeof(double))
                        + (long)(rms * nread * sizeof(double))
                        + (long)((time * (double)nread / 360. - 1.)
                                                * sizeof(double));

                /* Read the mean-velocity profile. */
                umprf[0] = 0.0; /* Set mean-velocity at the wall
                                                        to zero. */
                for(i = 1; i <= nsort || sortdist[i] <= 0.5*mdiam; i++){
                        /* Move to the entries corresponding
                                                to this phase angle. */
                        fseek(storefile,(rptr[i] + offset),0);

                        /* Read in the desired entry. */
                        fread(&umprf[i],sizeof(double),1,storefile);
                        if(diagnosis){
                                printf("velocity =%f\n", umprf[i]);
                                }
                        }

                /* Integrate profile at this time. */
                /* (Read the Oscillating Flow Experiment Log
                entry of 3/15/88 for details on the integration.) */
                umint = 0.0; /* Set integral to zero. */
                /* The integral excludes areas beyond the
                        center-line. */
                for(i = 1; i <= nsort; i++){
                        rim1 = 0.5 * mdiam - sortdist[i-1];
                        ri = 0.5 * mdiam - sortdist[i];
                        if(ri == rim1) {
                                continue;
                                }
                        else if(sortdist[i] <= 0.5 * mdiam) {
                                /* Integration between the far
                                   wall and the center-line. */
                                umint = umint + ((umprf[i-1] -
                                        umprf[i]) / (rim1 - ri)
                                        * (rim1*rim1*rim1/3.
                                                - ri*rim1*rim1/2.
                                                + ri*ri*ri/6.)
                                        + umprf[i]
                                        * 0.5 * (rim1*rim1 - ri*ri));
```

```
                                        }
                                    else if(rim1 > 0.0 && ri < 0.0) {
                                        /* Integration to the center-line. */
                                        /* ucl = interpolated
                                               centerline velocity */
                                        ucl = (umprf[i] - umprf[i-1]) /
                                               (sortdist[i] - sortdist[i-1])
                                               * (0.5*mdiam - sortdist[i-1])
                                             + umprf[i];
                                        umint = umint
                                               + ((umprf[i-1] - ucl) / rim1
                                                  * (rim1*rim1*rim1/3.)
                                               + ucl * 0.5 * rim1*rim1);
                                        break;
                                        }
                                    else {
                                        /* Exclude integration from
                                           center-line to the near wall. */
                                        continue;
                                        }
                                    }
                                if(diagnosis){
                                        printf("itime = %d, umint = %f\n",
                                        itime, umint);
                                        }

                                /* Calculate and store the mean-velocity
                                        averaged over the cross-section. */
                                fprintf(avefile, "%f\t%g\n", (angle =
                                        360. * (double)itime / (double)nread),
                                                                umint/aint);
                                if(diagnosis) printf("Writing to avefile complete.\n")
                                fprintf(idealfile, "%f\t%g\n",
                                        angle,
                                        pi * freq * mstroke *
                                        mbore*mbore / (mdiam*mdiam)
                                        * fabs(sin(pi*angle/180.)));
                                if(diagnosis) printf("Writing to files complete.\n");
                                }
                        fclose(avefile);
                        fclose(idealfile);
                        if(diagnosis) printf("Files closed. \n");
                        }

                if(fast) not_done = 0;
                }

/* Wake up the operator with a bell. */
for(i = 0; i < 10; i++) printf("%c", '\007');

the_end:;
/* End of the processing program "processx.c" */
}
```

VEL_REDTURB

```
/*    vel_redturb
      Converts (u,y) data profiles for turbulent-like portions of the cycle
      into (u+,y+) by iterating on the wall shear stress and y-offset.
      Results are stored in "upyp"
      The data is compared to curves which model the pressure-gradient effect.
      Results are stored in "upyppress"
      G. Friedman, 10/90  */

main()
{
#include <stdio.h>
#include <string.h>
#include <math.h>

#define N 200
#define N2 10
#define YPMAX 49
#define TUBE_D 1.5
#define TUBE_R 0.75
#define M 28.96
#define R 8315
#define CONV 0.0254
#define STRLNG 30
#define STRLNG2 100
#define PI 3.14159265
#define LIMIT 15

FILE    *profile,*datafile,*presfile;
int     a,b,i,j,k,l,m,inerr;
double  um[N],y[N],yorig[N],utau,uplus[N],yplus[N],ubm;
double  yovd,u,yoffset,dudt,pplus,angle,tmp,lnum,aplus,factor;
double  t_dry,p_atm,rho,nu,mu;
double  h,hover2,s,x,sf,half,dupdyp();
char    filename[],theta[4],resp[STRLNG],syst[STRLNG2];
char    syst2[STRLNG2];

printf("This program converts (u,y) data profiles into (uplus,yplus)\n");
printf("by iterating on entered values of cf (tau wall) and yoffset\n");
printf("** Turbulent profiles **\n\n");
printf("Input the crank position of the profile to be converted\n");
scanf("%s",theta);
printf("theta = %s\n",theta);
printf("Input the crank position of the profile to be converted\n");
scanf("%lf",&angle);
printf("angle = %f\n",angle);
strcpy(filename,"/usr/geoff/proc/prof");
strcpy(&filename[20],theta);
profile = fopen(filename,"r");
if(profile == NULL)
        {printf("This file does not exist.\n");
         exit();
        }
i = 1;
  printf("Note that um is corrected by 1.0205 for the temperature error\n");
do{
  fscanf(profile,"%lf %lf", &yovd. &u);
  um[i] = u * 1.0205;                          /* u is in m/sec */
  yorig[i] = CONV * yovd * TUBE_D;             /* convert y/d to meters */
  printf("i = %d y = %lf  u = %lf\n",i,yorig[i],um[i]);
  i++;
}
while (yovd < 0.5);
fclose(profile);
```

```
printf("\nEnter t_dry (degr.C) and p_atm (bar):\n");
scanf("%lf%lf", &t_dry, &p_atm);
printf("t_dry = %lf degr.C p_atm = %lf bar\n",t_dry,p_atm);
t_dry += 273.15;
p_atm *= 1.e+05;
rho = p_atm * M / R / t_dry;
printf("rho = %f\n",rho);
mu = 9.3277e-08 * t_dry  - 1.2248e-05;          /* viscosity at 1 atm */
mu *= (1.01325e+05 / p_atm);
printf("\nmu = %e\n",mu);
nu = mu / rho;
printf("nu = %e\n",nu);

/*      ****************************************       */
for(inerr = 1; inerr; ){
printf("\nInput the necessary y offset (in inches):");
scanf("%lf",&yoffset);
printf("yoffset = %f\n",yoffset);
yoffset *= CONV;
  for(k = 1; k <= i-1; k++){
        y[k] = yorig[k] - yoffset;
        /* printf("k = %d y = %lf  u = %lf\n",k,y[k],um[k]); */
        if(y[k] <= 0.0) y[k] = 0.0;
  }

printf("Input utau\n");
scanf("%lf",&utau);
printf("utau = %lf\n",utau);

datafile = fopen("upyp","w");
for(j = 1; j <= i-1; j++){
        uplus[j] = um[j] / utau;
        yplus[j] = y[j] * utau / nu;
  if(yplus[j] > 0.0)
/*      printf("j = %d yplus[j] = %f uplus[j] = %f\n",j,yplus[j],uplus[j]); */
        fprintf(datafile,"%f %f\n",yplus[j],uplus[j]);
}
fclose(datafile);

/*  uplus vs. yplus determination for effects of p-gradient  */
presfile = fopen("upyppress","w");
uplus[1] = 1.0;
b = 1;
fprintf(presfile,"%d %f\n",b,uplus[b]);
dudt = -8.725 * cos(angle * PI / 180.);
pplus = (nu * dudt) / pow(utau,3.);
if(angle <= 90.0)
        factor = 30.175;

else
        factor = 20.59;
aplus = 25.0 / ((factor * pplus) + 1.0);
if(aplus <= 0.0)
        aplus = 1000;
printf("aplus = %f\n",aplus);

h = 0.1;
hover2 = 0.05;
/*  loop to determine uplus vs. yplus  */
   for(m = 1;m <= YPMAX;m++){
        a = m;
        b = a + 1;
        s = 0.0;
        half = dupdyp((double)(a) + hover2,aplus);
        for(l = 1;l <= (N2-1);l++){
                x = (double)a + (double)l*h;
                s = s + dupdyp(x,aplus);
                half = half + dupdyp(x + hover2,aplus);
```

```
        }
        sf = (h/6.0)*(dupdyp((double)a,aplus) + 4.0*half + 2.0*s
                                        + dupdyp((double)b,aplus) );
        /* uplus determination */
        uplus[b] = sf + uplus[a];
        /* printf("yplus = %d uplus[%d] = %f\n",b,b,uplus[b]); */
        fprintf(presfile,"%d %f\n",b,uplus[b]);
    }
fclose(presfile);

printf("\nScreen plot of velocities desired? y or n\n");
scanf("%s",resp);
if(resp[0] =='y'){
        strcpy(syst,"graph -s -g 1 -x 1 .1 100 ");
        strcat(syst,"-y 0 35 5 < upyp | plot");
        system(syst);
        strcpy(syst2,"graph -s -g 1 -x 1 .1 100 ");
        strcat(syst2,"-y 0 35 5 < upyppress | plot");
        system(syst2);
        scanf("%s",resp);
        system("erase");
}

printf("Paper plot of velocities desired? y or n\n");
scanf("%s",resp);
if(resp[0] =='y'){
        strcpy(syst,"graph -s -g 1 -x 1 1 1000 ");
        strcat(syst,"-y 0 35 5 < upyp | plot -Thpib");
        system(syst);
        strcpy(syst2,"graph -s -g 0 -x 1 1 1000 ");
        strcat(syst2,"-y 0 35 5 < upyppress | plot -Thpib");
        system(syst2);
}

printf("aplus = %f\n",aplus);
printf("Input the bulk mean velocity\n");
scanf("%lf",&ubm);

printf("ubm = %f\n",ubm);
printf("utau = %lf\n",utau);
printf("cf = %f\n",pow((utau/ubm),2.) * 2.);

printf("\nContinue the iteration on yoffset and utau?\n");
scanf("%s",resp);
if(resp[0] == 'n') inerr = 0;
}
/*      ****************************************       */
printf("rho = %f\n",rho);
printf("utau = %f\n",utau);
printf("utau * utau = %f\n",utau * utau);
printf("tau wall = %f\n",utau * utau * rho);
printf("ubm = %f\n",ubm);
printf("cf = %f\n",pow((utau/ubm),2.) * 2.);
printf("aplus = %f\n",aplus);
printf("yoffset = %lf\n",yoffset);

/*  end  */
}
```

DUPDYP

```
#include <math.h>

#define KAPPA 0.41
#define square(x)       (x) * (x)

double  dupdyp(yplus,aplus)
double  yplus,aplus;
/*      This function determines duplus/dyplus for the given yplus */

{
double slope,arg1,arg2,arg3;
double damping;

        damping = 1.0 - 1.0/exp(yplus/aplus);

        /* Van Driest mixing length model with variable A+ */
                arg1 = square(KAPPA*yplus);
                arg2 = square(damping);
                arg3 = sqrt(1.0 + 4.0*arg1*arg2);
                slope = (-1.0 + arg3)/(2.0*arg1*arg2);

        return(slope);
}
```

VEL_REDLAM

```
/*  vel_redlam
    Converts (u,y) data profiles in the laminar-like portions of the cycle
    into (u+,y+) coordinates by iterating on the wall shear stress and the
    y-offset.  Results are stored in "upyp"
    Also generates a (u+,y+) curve based on a pressure-gradient influenced
    model for the Couette flow region.  Results are stored in "upyppress"
    G. Friedman, 10/90 */

main()
{
#include <stdio.h>
#include <string.h>
#include <math.h>

#define N 200
#define TUBE_D 1.5
#define TUBE_R 0.75
#define M 28.96
#define R 8315
#define CONV 0.0254
#define STRLNG 30
#define STRLNG2 100
#define PI 3.14159265
#define LIMIT 15

FILE    *profile,*datafile,*presfile;
int     i,j,k,l,inerr;
double  um[N],y[N],yorig[N],utau,uplus[N],yplus[N],ubm;
double  yovd,u,yoffset,dudt,const,angle,tmp,lnum;
double  t_dry,p_atm,rho,nu,mu;
char    filename[],theta[4],resp[STRLNG],syst[STRLNG2];
char    syst2[STRLNG2];

printf("This program converts (u,y) data profiles into (uplus,yplus)\n");
printf("by iterating on entered values of tau wall and y-offset\n");
printf("** Laminar profiles **\n");
printf("Input the crank position of the profile to be converted\n");
scanf("%s",theta);
printf("theta = %s\n",theta);
printf("Input the crank angle of the profile to be converted\n");
scanf("%lf",&angle);
printf("angle = %f\n",angle);
strcpy(filename,"/usr/geoff/proc/prof");
strcpy(&filename[20],theta);
profile = fopen(filename,"r");
if(profile == NULL)
        {printf("This file does not exist.\n");
        exit();
        }
i = 1;
printf("u corrected by 1.0205\n");
do{
  fscanf(profile,"%lf %lf", &yovd, &u);
  um[i] = 1.0205 * u;                          /* u is in m/sec */
  yorig[i] = CONV * yovd * TUBE_D;             /* convert y/d to meters */
  printf("i = %d y = %lf  u = %lf\n",i,yorig[i],um[i]);

  i++;
}
while (yovd < 0.5);                            /* ignore pts. beyond ctrlne */
fclose(profile);

printf("\nEnter t_dry (degr.C) and p_atm (bar):\n");
scanf("%lf%lf", &t_dry, &p_atm);
printf("t_dry = %lf degr.C p_atm = %lf bar\n",t_dry,p_atm);
t_dry += 273.15;
p_atm *= 1.e+05;
```

```
rho = p_atm * M / R / t_dry;
printf("rho = %f\n",rho);
mu = 9.3277e-08 * t_dry  - 1.2248e-05;          /* viscosity at 1 atm */
mu *= (1.01325e+05 / p_atm);
printf("\nmu = %e\n",mu);
nu = mu / rho;
printf("nu = %e\n",nu);

/*      ****************************************       */
for(inerr = 1; inerr; ){
printf("\nInput the necessary y offset (in inches):");
scanf("%lf",&yoffset);
yoffset *= CONV;
  for(k = 1; k <= i-1; k++){
        y[k] = yorig[k] - yoffset;
        /* printf("k = %d y = %lf  u = %lf\n",k,y[k],um[k]); */
        if(y[k] <= 0.0) y[k] = 0.0;
  }

printf("Input utau\n");
scanf("%lf",&utau);
printf("utau = %lf\n",utau);

datafile = fopen("upyp","w");
for(j = 1; j <= i-1; j++){
        uplus[j] = um[j] / utau;
        yplus[j] = y[j] * utau / nu;
        /* printf("yplus = %lf uplus = %lf um = %lf\n",yplus[j],uplus[j],um[j]);
        */
  if(yplus[j] > 0.0)
        fprintf(datafile,"%f %f\n",yplus[j],uplus[j]);
}
fclose(datafile);

presfile = fopen("upyppress","w");
dudt = 8.725 * cos(angle * PI / 180.);
const = (nu * dudt) / (2 * pow(utau,3.));

l = 0;
do{
l++;
uplus[l] = l - const * pow(1.0*l,2.);
        lnum = l;
/*      printf("\nyplus = %f uplus = %f",lnum,uplus[l]);        */
        fprintf(presfile,"%f %f\n",lnum,uplus[l]);
}
fclose(presfile);

printf("\ndudt = %f\n",dudt);
printf("const = %f\n",const);

printf("\nScreen plot of velocities desired? y or n\n");
scanf("%s",resp);
if(resp[0] =='y'){
        strcpy(syst,"graph -s -g 1 -x 1 .1 100 ");
        strcat(syst,"-y 0 35 5 < upyp | plot");
        system(syst);
        strcpy(syst2,"graph -s -g 1 -x 1 .1 100 ");
        strcat(syst2,"-y 0 35 5 < upyppress | plot");
        system(syst2);
        scanf("%s",resp);
        system("erase");
}

printf("Paper plot of velocities desired? y or n\n");
scanf("%s",resp);
if(resp[0] =='y'){
        strcpy(syst,"graph -s -g 1 -x 1 .1 100 ");
        strcat(syst,"-y 0 35 5 < upyp | plot -Thpib");
```

```
        system(syst);
        strcpy(syst2,"graph -s -g 0 -x 1 .1 100 ");
        strcat(syst2,"-y 0 35 5 < upyppress | plot -Thpib");
        system(syst2);
}

printf("Input the bulk mean velocity\n");
scanf("%lf",&ubm);
printf("ubm = %f\n",ubm);
printf("utau = %lf\n",utau);
printf("cf = %f\n",pow((utau/ubm),2.) * 2.);

printf("\nContinue the iteration on yoffset and utau?\n");
scanf("%s",resp);
if(resp[0] == 'n') inerr = 0;
}
/*      ****************************************        */
printf("rho = %f\n",rho);
printf("utau = %f\n",utau);
printf("utau * utau = %f\n",utau * utau);
printf("tau wall = %f\n",utau * utau * rho);
printf("ubm = %f\n",ubm);
printf("cf = %f\n",pow((utau/ubm),2.) * 2.);
}
```

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1992	Final Contractor Report

4. TITLE AND SUBTITLE

Fluid Mechanics Experiments in Oscillatory Flow
Volume II–Tabulated Data

5. FUNDING NUMBERS

WU–590–13–11
G–NAG3–598

6. AUTHOR(S)

J. Seume, G. Friedman, and T.W. Simon

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Minnesota
Dept. of Mechanical Engineering
Minneapolis, Minnesota 55455

8. PERFORMING ORGANIZATION REPORT NUMBER

None

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135–3191

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA CR–189128

11. SUPPLEMENTARY NOTES

Project Managers, Roy C. Tew and James E. Dudenhoefer, Power Technology Division, NASA Lewis Research Center, (216) 433–8471

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 34

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

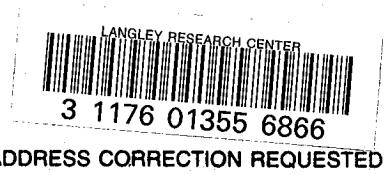
Results of a fluid mechanics measurement program in oscillating flow within a circular duct are presented. The program began with a survey of transition behavior over a range of oscillation frequency and magnitude and continued with a detailed study at a single operating point. Such measurements were made in support of Stirling engine development. Values of three dimensionless parameters, Re_{max}, Re_w, and A_R, embody the velocity amplitude, frequency of oscillation and mean fluid displacement of the cycle, respectively. Measurements were first made over a range of these parameters which included operating points of all Stirling engines. Next, a case was studied with values of these parameters that are representative of the heat exchanger tubes in the heater section of NASA's Stirling cycle Space Power Research Engine (SPRE). Measurements were taken of the axial and radial components of ensemble-averaged velocity and rms-velocity fluctuation and the dominant Reynolds shear stress, at various radial positions for each of four axial stations. In each run, transition from laminar to turbulent flow, and its reverse, were identified and sufficient data was gathered to propose the transition mechanism. Models of laminar and turbulent boundary layers were used to process the data into wall coordinates and to evaluate skin friction coefficients. Such data aids in validating computational models and is useful in comparing oscillatory flow characteristics to those of fully-developed steady flow. Data were taken with a contoured entry to each end of the test section and with flush square inlets so that the effects of test section inlet geometry on transition and turbulence are documented. The following is presented in two volumes. Volume I contains the text of the report including figures and supporting appendices. Volume II contains data reduction program listings and tabulated data (including its graphical presentation).

14. SUBJECT TERMS	15. NUMBER OF PAGES
Fluid mechanics; Oscillating flow; Stirling engines	184
	16. PRICE CODE: A09

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

National Aeronautics and
Space Administration

Lewis Research Center
Cleveland, Ohio 44135



NASA